NPS-SP-95-003

NAVAL POSTGRADUATE SCHOOL
Monterey, California

DEVELOPMENT OF A COMPUTER-CONTROLLED INSTRUMENTATION FOR A THERMAL VACUUM CHAMBER

by

Michael Hengst

September 1995

Prepared for: Unversitaet der Bundeswehr Muenchen
85579 Neubiberg, Germany

NAVAL POSTGRADUATE SCHOOL
Monterey, California

Rear Admiral M.J. Evans
Superintendent

R. Elster
Provost

This report was prepared for Unversitaet der Bundeswehr Muenchen, 85579 Neubiberg, Germany.

The report was prepared by:

Universitaet der Bundeswehr Muenchen
GERMANY

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE September 1995	3. REPORT TYPE Technical Report

4. TITLE AND SUBTITLE
Development of a Computer-Controlled Instrumentation for a Thermal Vacuum Chamber

5. FUNDING NUMBERS

6. AUTHOR(S) Michael Hengst

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(ES)

Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

NPS-SP-95-003

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Unversitaet der Bunderwehr Muenchen
85579 Neubiberg , GERMANY

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the US Government.

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)
In this thesis, a Tenney Space jr. Thermal Vacuum Chamber (TVAC) was modified in a way that allows complete control of the system and the performance of tests under space specifications by computer. To achieve this, both computer hardware and software were installed and modified and additional mechanical connectors were designed and installed. A large series of tests throughout all phases of this project was performed to test the equipment, to learn about the TVAC controls and to prove that the system is fully operational after the changes.

14. SUBJECT TERMS
Development of a Computer-Controlled Instrumentation for a Thermal Vacuum Chamber

16. PRICE CODE

15. NUMBER OF PAGES

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

ABSTRACT

This thesis documents the development of a computer instrumentation system capable of controlling a Tenney® Space jr. thermal-vacuum chamber (TVAC).
The project was performed to achieve more precise temperature control of the chamber through the use of an integrated computer and software solution rather than using the TempTenn controller which was originally installed in the TVAC. Three objectives were to be achieved: Automated data acquisition, automated temperature control and ease of use.

The system uses a 486DX-4 100MHz personal computer with the National Instruments® SCXI hardware and LabVIEW software to perform data acquisition, to control the TVAC functions and to present a Graphical User Interface (GUI). Temperature measurements are acquired by the system to indicate the environment as well as to control the test. The automated temperature control allows the system to maintain a target temperature from -65°C to 175°C with an accuracy of ±3 °C. Furthermore the system allows to store data to a disk. A very convenient GUI was provided for ease-of-use and visual presentation of the data acquisition.

This project was designed and built for environmental testing of components and subsystem modules at pressures as low as 10^{-7} torr. The modified thermal-vacuum chamber will be especially useful in the forthcoming tests related to the PANSAT small satellite project of the Space Systems Academic Group (SSAG) at the Naval Postgraduate School.

Table of Contents

I. INTRODUCTION

A. SCOPE OF THIS THESIS

The purpose of this thesis is to develop a computer-controlled thermal vacuum chamber testing device .

A manually controlled Thermal Vacuum Chamber (TVAC) of the type "SPACE jr." by Tenny Industries was modified to allow complete control by a PC, providing temperature and pressure measurements, thermocouple readings, data storage and more. This thesis describes the complete list of changes which had to be undertaken, covering many fields of engineering:

In order to change the existing TVAC, National Instruments SCXI hardware devices were installed for data acquisition. These tasks are described in chapter III.

Several mechanical and electrical changes were made to the TVAC. This resulted in several design tasks, which had to be performed by means of computer aided design (CAD). The description of these tasks can be found in chapter IV.

The biggest part of the project was to do program development and software design under LabVIEW. The goal was to achieve precise control. Chapter V refers to the LabVIEW programming, explaining the strategy that was followed.

Several tests were conducted to test the function of the TVAC, during and after the modifications. This was the most time-consuming part of this thesis. The different steps of the procedure and some test results are presented in Chapter VI .

B. THERMAL VACUUM TESTING

Thermal testing of spacecraft components is a major design topic, as it is of vital importance to make sure that all subsystems and components will function in the space

thermal environment. The thermodynamic state of a spacecraft is determined by heat irradiation from the sun, radiation between the spacecraft's surface and the internal components and heat transfer (conduction and radiation) through internal paths. There is no convection in the vacuum of space, so heat is transferred within the spacecraft by conduction and radiation only.

Throughout all aspects of a spacecraft's mission, which normally covers a wide thermal spectrum, the thermal condition has to be maintained within a specified limit or range, which can only be achieved by a proper thermal control design, be it active or passive. This control has to be accomplished before launch, during ascent into orbit, after the heat shield is jettisoned (if the launch vehicle is a rocket) and throughout all transfers to a final orbit or trajectory.

Since testing under mission conditions allows much more precise statements on the components' function and reliability than simple calculation, space environment conditions have to be simulated. Thermal vacuum testing provides simulation of a mission's thermal spectrum and heat dissipation in a spacecraft, following a pre-defined thermal pattern like the one that is expected throughout the spacecraft's mission. The thermal condition of every single part of a subsystem can be measured by means of thermocouples, and functional tests of the whole system can be carried out under space conditions. Even failures due to material weakness or flawed structural design can be discovered and problems solved before the launch, which can avoid a possible loss of a spacecraft during the mission.

Thermal vacuum testing on earth cannot simulate all influences the spacecraft will be exposed to in space, such as changes in gravity or solar radiation which darkens surfaces and thus changes their thermal properties.

Spacecraft integration and test are the final steps in the development of a spacecraft before integration to the launch vehicle. Testing normally includes optical and mechanical alignment, determination of mass properties and magnetic and electromagnetic fields, vibration tests of subsystems and the integrated system, electrical baseline tests under ambient conditions, thermal shock tests of components and thermal

vacuum tests of the systems and subsystems. This thesis will only deal with the last topic, thermal vacuum testing.

Thermal vacuum testing is performed by means of a Thermal Vacuum Chamber (TVAC). Testing consists of thermal cycling under a vacuum. One thermal cycle is performed with dwell times of 60 minutes for each temperature extreme. Normally, maximum and minimum temperatures shall be 100°C and -40°C. The average temperature rise and fall rate shall be 1°C per minute.[1] Testing is performed with pressure below $1.0*10^{-6}$ torr, but pressure values can fall to a minimum of $3.0 * 10^{-7}$ torr, depending on the temperature.

[1] SPACE SYSTEM ACADEMIC GROUP, Solar Panel Thermal Vacuum Test Plan

II. BACKGROUND

A. PETITE AMATEUR NAVY SATELLITE

The Space Systems Academic Group (SSAG) at the Naval Postgraduate School in Monterey, California, supports Space Systems Engineering and Space Systems Operations Curricula. In 1989, the SSAG initiated the design of a small, low-cost satellite for digital, spread spectrum communication using the amateur frequency band.

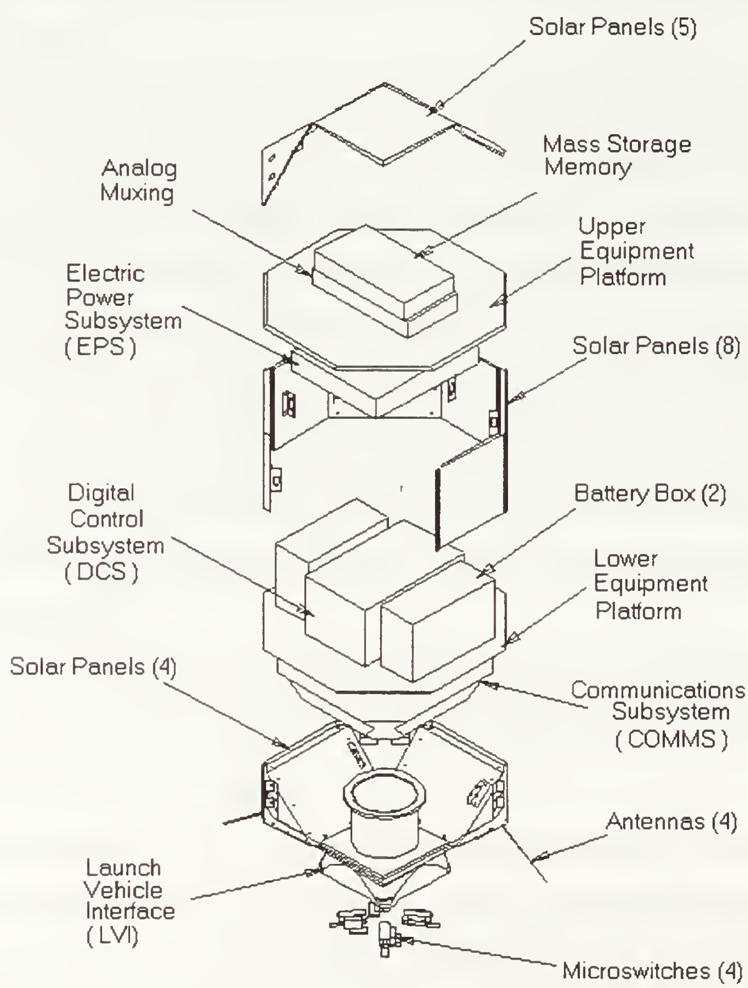


Fig. 1 : Pansat exploded view

A major aim in developing this satellite is to provide educational opportunities for officer students in the Space Systems curricula. Many theses on practical research topics have been written covering all possible aspects of PANSAT. Some example topics are :

System design, system integration, hardware and software development, ground-station design, programming, implementation of communication, testing, and operation studies. Thus, the project allows students, engineers and technicians to take part in developing a space qualified satellite through hands-on work.

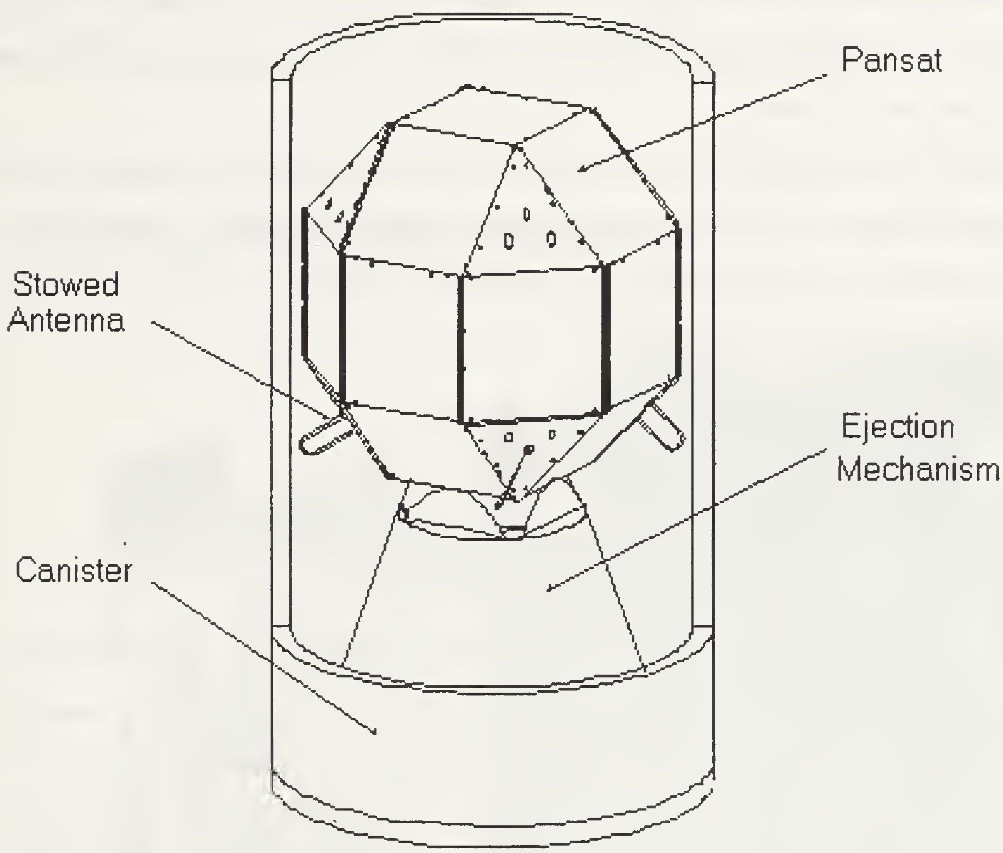


Fig. 2 : Pansat in Get Away Special (GAS) canister

PANSAT has been designed as a low-earth orbit, free-tumbling satellite in an orbit of 480 km and approximately 51.6 degrees inclination. The expected time of operation is 2 years. PANSAT consist of 4 subsystems : Structure, Electrical Power, Communication, Digital Control (computer).

The outer structure is made of aluminum 6061-AT approximating a sphere with a diameter of about 19 inches. The Communications Subsystem uses an amateur packet radio, which allows error-free digital data exchange at high speed, using a spread-spectrum, binary phase shift keying-modulation (BPSK).

This will allow communication with multiple users at a time during a communication window of about 6 minutes per orbit . While connected, the user may send and receive mail that is stored on-board PANSAT, upload and download files, and also read spacecraft telemetry.

PANSAT is scheduled to be launched from the Space Shuttle by means of a Get Away Special Canister (GAS) on a Shuttle Mission to the MIR-Station. This will provide a higher orbit and thus a longer communication window.

B. SPACE ENVIRONMENT

A system in space is influenced and acted upon by various environmental factors, which have to be taken into account when designing and operating a spacecraft. The vacuum of space is one important factor and so is the temperature in space, which can be very low or very high and thus very different to that on earth. As a result, mechanical and electrical subsystems must include these considerations. This means that expected temperature limits throughout the operation have to be taken into account. Typical temperature design limits for spacecraft systems are 0 to 50°C for electronics and 0 to 20°C for batteries.

A spacecraft is also exposed to gravitational, electrical and magnetic fields and a wide spectrum of radiation, ranging from particles to waves. The amount of influence of these variables is dependent on the specific orbit, the orbit's inclination and altitude.

These parameters affect the thermal and material properties of a spacecraft (for example the shuttle glow phenomenon which is caused by atomic oxygen mitigating spacecraft charging). Radiation is an especially endangering factor for a spacecraft's mission.

Another difficulty is that the environment changes throughout a mission, be it through the different stages of launch to orbit or in space, when a spacecraft transits through the shadow of a planet experiencing sudden extremes in thermal conditions. These effects have to be taken into account in selecting a proper orbit.

C. THERMAL VACUUM CHAMBER

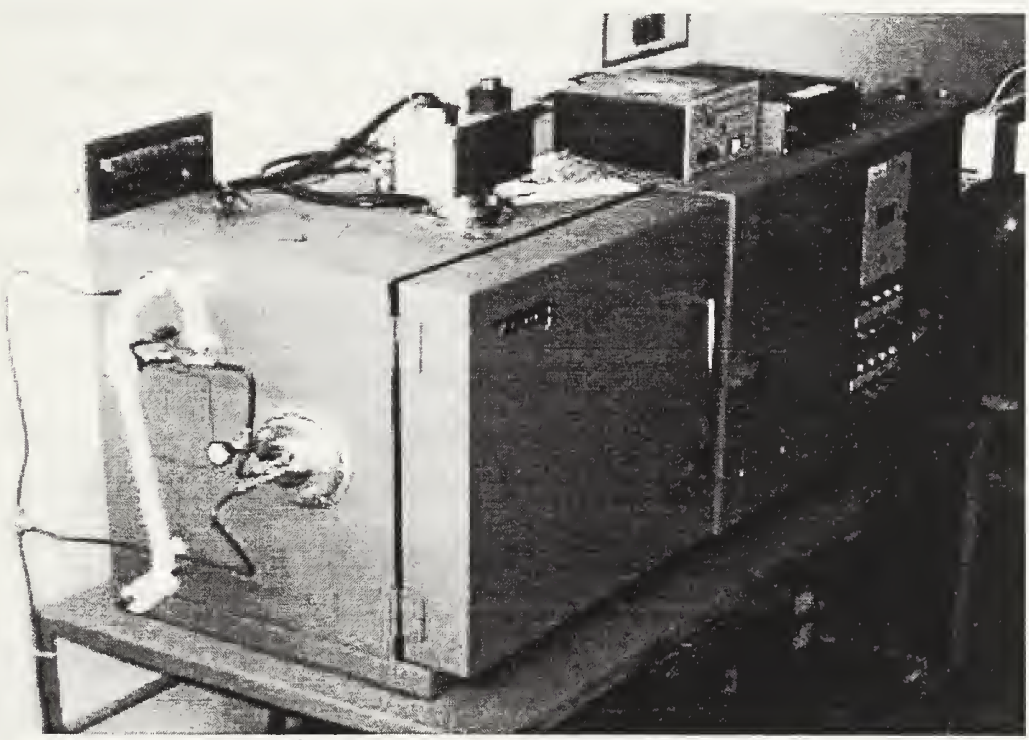


Fig. 3: *Photo of the Space jr. Thermal Vacuum Chamber*

The Thermal Vacuum Chamber involved in this project is a product of Tenney Industries, Model Sjr. (SPACE JR.). It is a precision, bench model, thermal vacuum space simulator for the development and testing of small components in a simulated space environment.

Space Jr. Specifications :

Work Space :	14" Diameter x 12" Deep (36 cm x 31 cm)
Vacuum Instrumentation:	Hot filament ionization gauge and thermocouple gauge
Temp.Instrumentation :	Solid state proportioning controller
Temperature Readout:	Thermocouple wire for use with customer supplied readout
Refrigeration:	Cascade System with air-cooled condenser
Two Cold Traps:	Mechanically refrigerated and LN_2
LN_2 Consumption:	15 Pounds/Hr.(for cold trap only)
Overall Dimensions:	56"W x 26"H x 24"D (142 cm x 66 cm x 61 cm)
Insulation:	5" Fiberglass
Power Requirements:	220 Volts, 1 Phase, 60 Hz, 30 Amps
Shipping Weight:	750 lbs. (240 kg) approx.

Temperature Performance (Manufactor's Specifications) :

Pulldown :	24°C to -45°C in approximately 45 minutes
	24°C to -65°C in approximately 65 minutes
	24°C to -73°C in approximately 90 minutes
Heatup :	24°C to 121 °C in approximately 30 minutes
	24 °C to 177°C in approximately 60 minutes

Control Tolerance : ± 0.56 °C (at the extremes)

Vacuum Performance (in the shipping state, before the pumps were exchanged) :

	Temperature	Vacuum	Time to evacuate (approx.)
Fore Pump	24°C to -73°C	100 microns	1 hour
Fore Pump	24 °C to 177°C	150 microns	1 hour
Diffusion Pump (with mechanical trap)	24°C to 177°C	1.0×10^{-5} torr.	6 hours
Diffusion Pump (with mechanical trap)	24°C to -73°C	3.0×10^{-6} torr.	2 hours
Diffusion Pump: (with LN_2 trap)	-73°C	7.5×10^{-8} torr.	10 hours

Temperature conditioning is accomplished by means of refrigerating tubing and electric heaters thermally bonded to the chamber walls. A hermetically sealed cascade refrigeration system is used to provide sub-ambient temperatures.

The test chamber is constructed of stainless steel to provide good heat transfer and minimum leak rate. The interior is blackened to permit high radiant energy transfer. The chamber is enclosed in a hermetically sealed, insulated, sheet metal container. Access to the vacuum chamber is through a thermal door and a separate, free floating, positive seating vacuum door.

The vacuum system consists of a turbo molecular pump and a rotary mechanical pump. A panel mounted valve is provided for venting and backfilling with gaseous N_2. A high capacity, liquid nitrogen cooled baffle is also provided.

A solid-state proportioning controller is provided to control the test chamber wall temperature. A hot filament ionization gauge with dual tungsten elements is provided to monitor chamber pressure. It is readable from 1×10^{-3} to 2×10^{-3} to 2×10^{-9} torr. A thermocouple gauge is provided for reading fore pressure.

The unit is equipped with two accessory feed-through ports (one 2" and one 3") and with a special trap for auxiliary fore pump capacity.[Ref. 8]

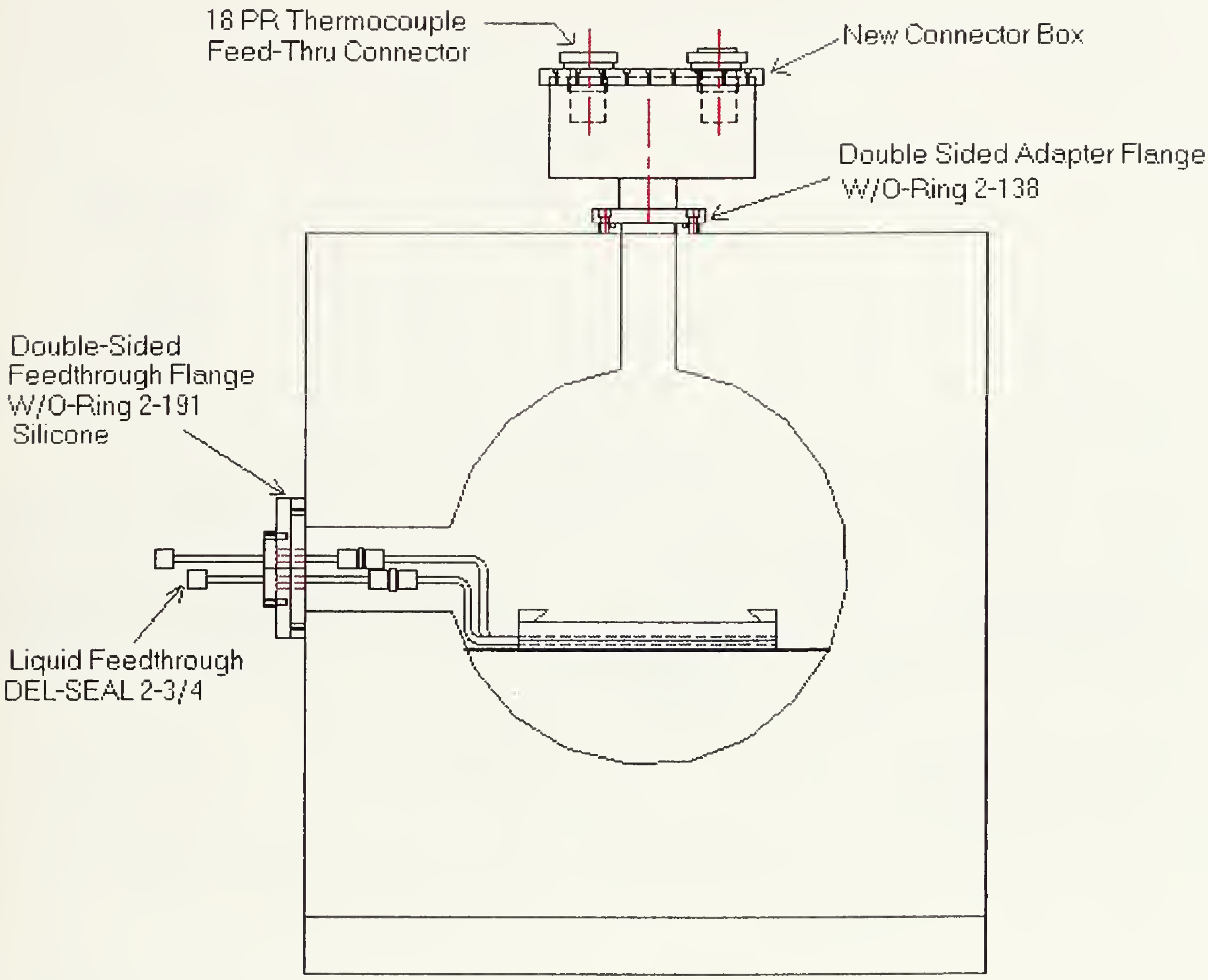


***Fig. 4** : View of the chamber with the new connector box*

Before this project was started, a modification of the shipping state was made. The Varian fore pump was replaced with an Edwards pump and the Boekel Hyvac pump with a VARIAN Turbo-V200 Pump. This configuration is cleaner and will reach a lower minimum pressure.

The controller of the V200-Pump was mounted on top of the chassis, next to the vacuum gauge controller. As a result of these modifications the actual power of the system varies from the values stated by Tenney Industries, but no exact values are known.

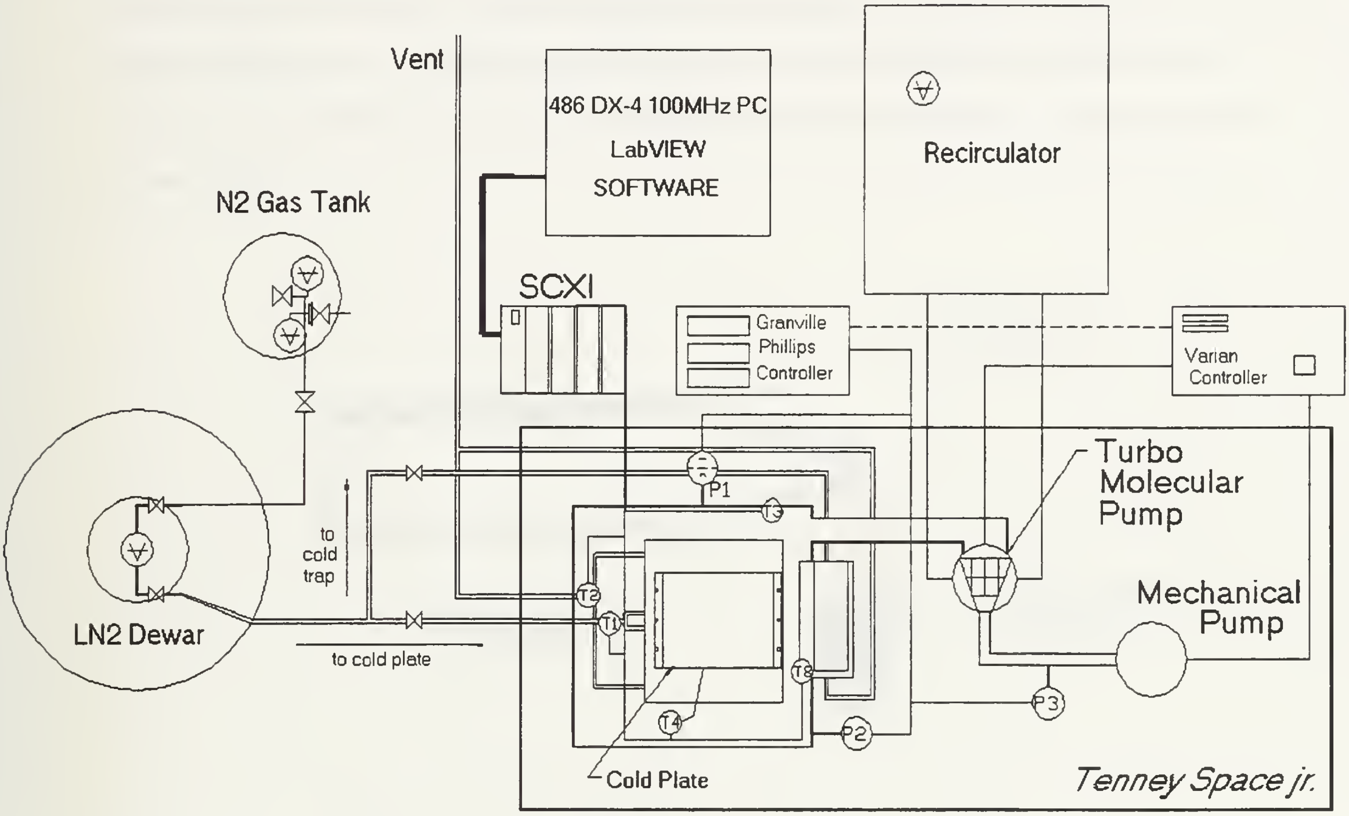


Fig. 5 : Space jr. Thermal Vacuum Chamber Block Diagram

D. THERMOCOUPLE - MEASUREMENTS

Probably the most important means of measuring temperatures is the thermocouple. Since thermocouples are relatively cheap but can deliver exact thermal values over wide temperature ranges, they are one of the most frequently used temperature transducers. They are used by the Tenney thermal vacuum chamber which is the topic of this thesis.

How do thermocouples work ? When two wires, which consist of dissimilar material, are put together on both ends and one junction is exposed to a heat source, a continuously flowing current can be measured in this circuit.

This effect is called 'The Seebeck Effect" after Thomas Seebeck, who discovered the phenomenon in 1821. If the wires are only connected at the end where the heat exposure takes place, a voltage can be measured at the open end, the so-called Seebeck Voltage. It is a function of the junction temperature and the composition of the material of the two wires.

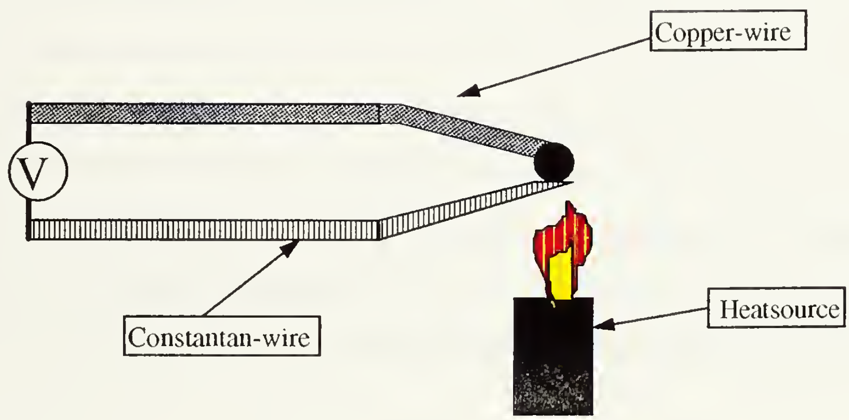


Fig. 5 : Thermocouple Type T (Copper-Constantan)

Although the Seebeck Voltage is non-linear with respect to temperature, it can be approximated to be linearly proportional to temperature for small changes in temperature:

$$\Delta V = \alpha \Delta T$$

where α, the Seebeck coefficient, is the constant of proportionality, ΔT is the change in Temperature and ΔV is the change in the measured voltage .

The Seebeck coefficient α is nonlinear over the measured temperature ranges, causing a nonlinear output voltage. α is a function of the material composition, and thus there exist several types of thermocouples. They are designated by capital letters that indicate their composition and thus their Seebeck coefficient and the temperature range which they can cover. The convention for the thermocouple designation was made by the American National Standards Institute (ANSI).

Measuring a Seebeck Voltage, however, is not as easy as it might appear, because connecting the thermocouple leads to a measurement system generates additional thermoelectric circuits, which influence the voltage .

This means that connecting the thermocouple to a voltmeter or a Data-Acquisition Board creates two new junctions which have their own Seebeck coefficient and therefore generate their own thermoelectric voltage proportional to the temperature at the connector terminals. Thus the voltage contribution of the junction where the temperature is to be measured can only be determined, if the temperatures and the voltage-to-temperature-relationship of the newly created junctions are well known. Then the voltages generated by these two junctions have to be subtracted from the total voltage read by the DAQ-Board.

To allow an accurate temperature reading it is necessary to have a reference temperature for the reading : The cold-junction temperature. This name originates from the former method by which the reference temperature was achieved, which was holding the junction at 0°C in an ice-bath. When the temperature of the measurement junction and the reference junction are the same, the net voltage is zero.

In Figure 7 the junctions between the Constantan wires and the voltmeter connections are kept isothermal which has the effect that the voltages generated at these junctions are equal and opposing. Therefore no further net voltage error is added by these connections.

In this case one will obtain the following result :

The output is positive if the junction to be measured is above 0°C (which is the reference junction temperature), and negative if the temperature is below zero.

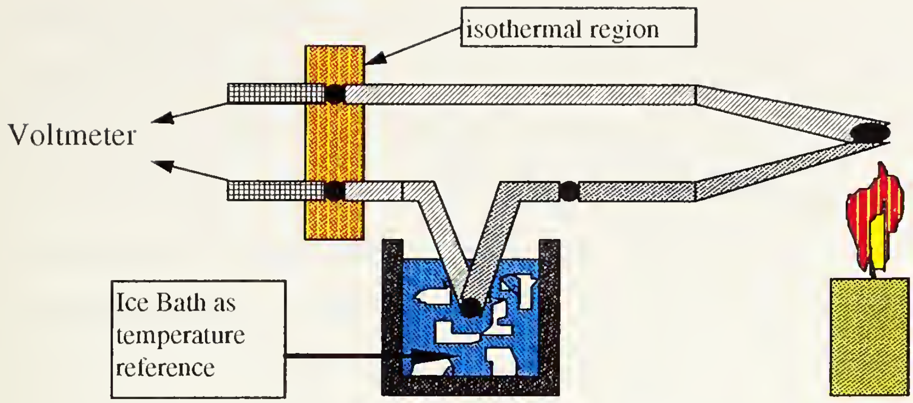


Fig. 7 : Ice Bath as Reference Junction Temperature

As it is not always convenient to use an ice-bath, another, more practical approach to the problem can be made, the so-called "Cold-Junction Compensation".

The temperature of the reference junction is measured with a direct-reading temperature-sensor, and all other contributions to the thermoelectric voltage, so-called parasitic voltages, are subtracted. Only a simple computation is needed. The assumptions leading to this simplification are presented next:

The so-called "Thermocouple Law of Intermediate Metals" states that inserting any type of wire into a thermocouple circuit has no effect on the output, as long as both ends of that wire are isothermal.

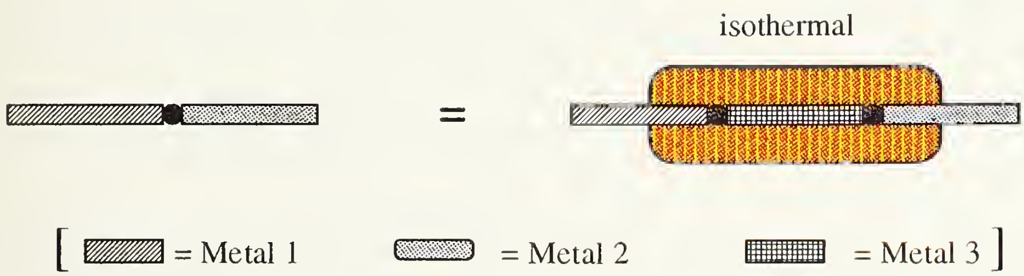


Fig. 8 : Thermocouple Law of Intermediate Materials

Furthermore makink use of the fact that two junctions of the same type, which occur in opposite directions in the same isothermal region, oppose their generated voltages and therefore effect a net voltage of 0 volts (see Fig. 6). Considering these rules, the schematics in Fig.9 can be created. In this circuit, J1 and J2 are of same type and same temperature, and as they appear in opposing directions, they do not generate any additional voltages. Junctions J3 and $J_{measurement}$ are the only junctions who give a contribution to the voltage measured by the voltmeter.
They are of the same type but at different temperatures and, therefore, generate different voltage contributions to the total voltage measured by the Voltmeter.

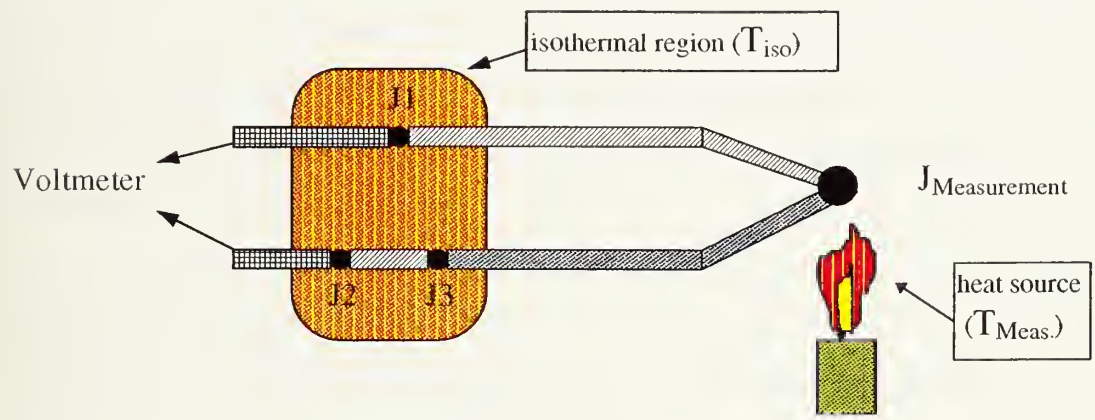


Fig. 9 : *Compensation Circuit*

As a result, the following important equation is obtained:

$$V_{Voltmeter} = V_{J3}\ (\text{at } T_{iso}) + V_{measurement}\ (\text{at } T_{meas.}) \tag{1}$$

where $V_{Voltmeter}$ is the voltage read by the acquisition system, V_{J3} is the voltage generated at Junction J3 and $V_{Measurement}$ is the voltage created by the heat source at the end of the thermocouple-wire. The equation is valid only in case that $V_{Measurement}$ and V_{J3} are functions of temperature **relative to the same reference temperature**.

Since J3 and $J_{measurement}$ are of the same type and in opposing directions, they would generate voltages opposing each other to zero if they were at the same temperature, so that they can both be called J_{TC}, the voltage generated by them V_{TC}. If we rename T_{iso} as T_{ref}, and $T_{meas.}$ as T_{TC} we obtain the following :

$$V_{voltmeter} = V_{TC}\ (\text{at } T_{TC}) + V_{TC}\ (\text{at } T_{ref}) \qquad (2)$$

This equation indicates that, if the voltage-to-temperature relationship of the thermocouple and the reference-junction temperature are known, the temperature at the measuring thermocouple can be determined.

In general, there are two kinds of cold-junction reference methods which can be used : hardware compensation and software compensation. Both techniques require that the temperature at the reference junction be known or measured with a direct-reading sensor, which is a sensor whose output depends only on the temperature of the measurement point.

a) hardware compensation

   A variable voltage source is inserted into the circuit to cancel the parasitic thermoelectric voltages. The variable voltage source generates a compensation voltage according to the ambient temperature, and thus adds the correct voltage to cancel the unwanted thermoelectric signals.

b) software compensation

   After a direct-reading sensor measures the refernce-junction temperature, software can add the appropriate voltage value to the measured voltage to eliminate the parasitic thermocouple effects.

The SCXI Instrumentation allows both software and harware compensation, but the technique used by my program is hardware compensation.

III. COMPUTER-CONTROLLED INSTRUMENTATION

A. LEVEL OF AUTOMATION

Before starting the work on the TENNEY Space Jr. Thermal Vacuum Chamber the team of the Space System Academic Group (SSAG) discussed the desired level of automation of the TVAC which was to be achieved by this thesis project.

The question was if the computer should be able to control the TVAC totally, including the inlet valve for liquid nitrogen, or if the level of automation should be restricted to temperature control. Being concerned that a breakdown of the power supply could create a possible failure in the nitrogen inlet, which could result in a fatal crash of the TVAC, it was decided to have the computer do temperature control only. Creating a vacuum in the chamber and inserting liquid nitrogen were decided to be done manually as before.

This means that the switches, “heat”, “doorheat”, “ambient cooling”, “sub-zero cooling” and “cold trap”, which can be found on the front panel of the TVAC, had to be modified to allow computer control. Furthermore the computer is supposed to take temperature and pressure measurements and to save the data to a hard disk in a way that the data can be imported into a Microsoft Excel Datafile and thus presented in a chart. To achieve this, 18 thermocouples had to be connected in and around the TVAC and the hardware which is necessary to read their outputs had to be installed . Furthermore a connection between the computer and a pressure-reading device had to be established.

B. HARDWARE-DEVICES

1. SCXI-System

The hardware used in this case is a NATIONAL INSTRUMENTS® SCXI system. SCXI stands for Signal Conditioning eXtensions for Instrumentation and is a trademark of NATIONAL INSTRUMENTS®. The SCXI 1000 chassis is a 4-slot chassis housing the following modules:

- SCXI 1100 : 32-channel differential multiplexer / amplifier module, analog input
- SCXI 1120 : 8-channel isolated analog input module
- an empty slot
- SCXI 1161 : 8-channel power relay module, digital input/output

The SCXI-chassis is connected to a **AT-MIO 16E-10** - Data Acquisition Board (DAQ-Board) which is plugged into one of the slots in the 486 DX-4 100 MHz Personal Computer.

The AT-MIO 16E-10 multifunction board can control single-chassis and multi-chassis SCXI systems consisting of any combination of analog and digital SCXI modules.

The SCXI 1000 chassis consists of an analog bus, a digital bus and a chassis controller that regulates bus operation.

The DAQ-board communicates with the SCXI chassis and the controller via the digital bus , the analog bus transfers analog signals directly to the computer. Thus analog signals are conditioned and passed back to the plug-in board for acquisition directly into PC memory.

The plug-in board is connected to only one module in the SCXI chassis, which is the SCXI 1100 module in this case. Using SCXI multiplexing, the DAQ-board in the computer can acquire thousands of conditioned analog signals using multiple SCXI boards.

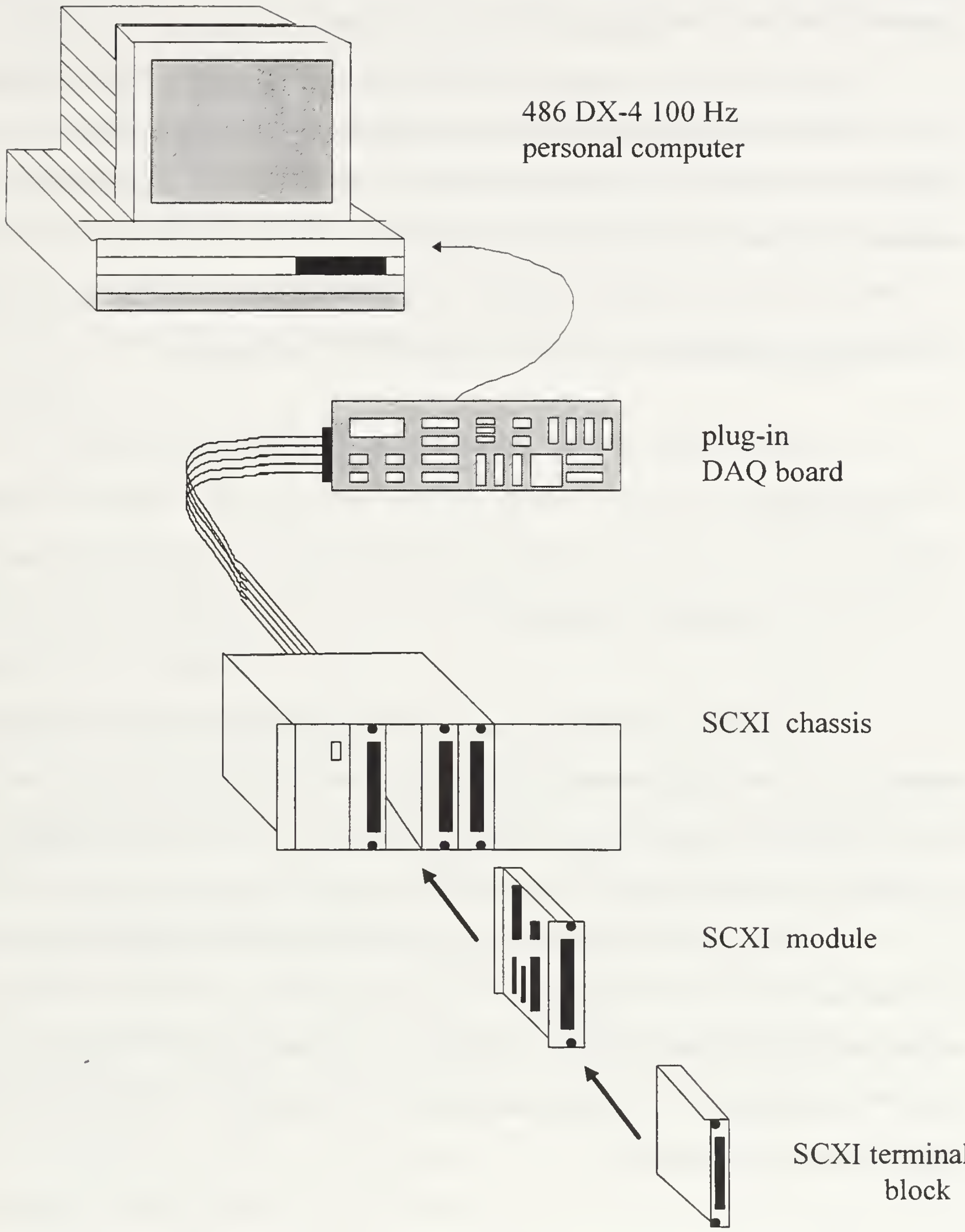


Fig. 10 : SCXI System Components

Signals are connected to the modules by using terminal blocks or connector-and-shell assemblies, which are plugged into the front of the SCXI modules.

Terminal blocks are rugged shielded termination assemblies with a number of screw terminals, to which signal wires can be connected, and a strain relief to prevent the connections from being inadvertently loosened . This allows the use of different transducers without removing the whole chassis.

The SCXI-1100 module carries a SCXI-1300 terminal block and a SCXI-1320 terminal block is plugged into the front of the SCXI-1120 module.

2. SLOT 1

a. SCXI-1100 Module

The SCXI-1100 module is a 32-channel differential-input multiplexer. It contains an onboard programmable-gain instrumentation amplifier (PGIA). The module has 32 differential voltage or current inputs, on-board jumper-selectable lowpass filters and software-selectable gains. The analog input range is between +10V and -10V.

The SCXI-1100 works only in multiplexed mode and drives two analog input channels (channel 0 and channel 1) on the DAQ-board. The SCXI-1100 is the module in the SCXI-1000 chassis that is directly connected to the AT-MIO 16E-10 board. It is responsible for communication between the DAQ-board and the other modules in the SCXI system.

b. SCXI-1300 Terminal Block

The SCXI-1300 terminal block is plugged into the front of the SCXI-1100 module. The SCXI-1300 terminal block provides 32 screw terminals for signal cables. In this application, 18 thermocouples are connected to the first 18 screw terminals.

The SCXI-1300 has an onboard temperature sensor for measuring ambient temperature inside the terminal block. This temperature can be used for cold-junction compensation when measuring temperatures with thermocouples.

One has to differentiate between ground-referenced sources, (which simply means that the source is grounded) and floating sources. Here, the thermocouples are of the floating source type. The thermocouples used for this project are Omega SA1-T-type precision fine bar wire thermocouples. The thermocouple wires have a diameter of DIA: 0.010 - 36 inch. They are Teflon insulated and provide a temperature range of -200°C to +350°C .

Since thermocouples inevitably pick up some amount of noise, it is recommended to sample them at a higher rate than needed. In this case, their values are measured and more than 30 points are averaged per reading. The signal of the on-board temperature sensor is multiplexed and sent to channel 0 on the DAQ board , along with other signals from the module.

3. SLOT 2

a. SCXI-1120 Module

The SCXI-1120 module is an 8-channel isolated amplifier / analog input module with the following features: eight isolated input channels, an analog input range of +5 to -5 V, 240 Vrms common-mode voltage, 1,500 Vrms breakdown isolation per channel, jumper-selectable lowpass filter and gain.

In general, the SCXI-1120 is a module for signal conditioning of thermocouples or (milli)volt sources and 4 to 20 mA current sources. In this case the module is used to read the temperature and the pressure of the Space Jr. TVAC-chamber. The module carries a SCXI-1320 terminal block used to connect it to the original temperature controller of the chamber, called TempTenn. The SCXI-1120 module is plugged into the analog and digital bus backplane of the SCXI chassis. The backplane

connects the module to the SCXI-1100 module. The SCXI-1120 module is operated in multiplexed mode.

b. SCXI-1320 Terminal Block

The SCXI-1320 is a shielded board supporting the connection of input signals to the SCXI-1120 module. It has 18 screw terminals for easy connection. Eight pairs of those are for signal connection to the eight input channels of the SCXI-1120 (of which only one is connected in this application), and one pair of screw terminals connect to the chassis ground. The terminal block has a temperature sensor for cold-junction compensation, which is not used in this case.

4. SLOT 4

a. SCXI-1161 Module

The SCXI 1161 module is a digital module which can switch and control eight isolated single-pole double-throw (SPDT) relays. Its major components are the SCXI bus connector, the digital interface, the digital control circuit, and the relay channels. The SCXI-1161 is connected to the SCXI 1100 through the backplane of the chassis, via the SCXI-1100 module to the DAQ-board. The SCXI-1161 has 24 screw terminals for easy signal connection. In this application, 6 relays are connected to the onboard screw terminals of the module :

#1 : cold trap	#4 : heat
#2 : sub-zero cooling	#5 : doorheat
#3 : ambient cooling	#6 : bypass mode on/off

The relays #1-5 are to take over the control features when the TVAC is run in the computer-controlled mode. Relay #6 allows for heating up the TVAC without using or being dependent on the installed TempTenn controller .

C. SOFTWARE

The software which was chosen to control the Space Jr. TVAC-chamber is LabVIEW® from NATIONAL INSTRUMENTS®. LabVIEW is a program development application which is run under Microsoft Windows® 3.11.

LabVIEW uses a graphical programming language called "G", by which programs can be created in block diagram form. LabVIEW is shipped with various libraries and functions for any programming task, which makes it applicable for many different purposes. One reason why this program was chosen for this task is that LabVIEW includes libraries for data acquisition, serial instrument control, data analysis, data presentation and data storage. It also includes tools like breakpoint-setting, single-stepping and animated execution for more comfortable debugging. Thus LabVIEW provides a very comfortable means for program development.

LabVIEW programs are called **virtual instruments (VIs)** because their appearance and operation can imitate actual instruments. VIs are like functions in conventional test-based languages and they can be given parameters from higher level VIs.

Using the "front-panel" on the monitor, which can be configured like a physical instrument, VIs can be interactively used by entering data via a keyboard or a mouse. The VIs are given instructions from block diagrams, which can be constructed in G and which are the source code for the VI . Thus a program is written in a pictorial way. The block diagrams are constructed by wiring together objects that send or receive data, perform specific functions, and control the flow of execution.

Another feature of LabVIEW is the concept of modular programming, which means that an application can be divided into a series of tasks, so that complicated applications become a series of multiple simple tasks. Those VIs within other VIs are called subVIs. Data can be passed between VIs by a graphical parameter list called connector. Using this, one can build top-level VIs that consist of various subVIs which are suitable to accomplish various subtasks.

In contrast to a conventional program, the principle of LabVIEW execution is not instruction flow (instructions are executed in the sequence they are written) but data flow. Stated simply, a node (terminal, control, etc.) executes only when data arrives at **all** its input terminals. The node supplies data to all its output terminals immediately when it finishes executing.

D. ADDRESSING THE HARDWARE

The different modules in the SCXI chassis have different addresses, but mostly use the same onboard channel. The SCXI hardware devices are addressed by two inputs : the *channel* and the *device-number*. They are normally entered into control windows in the VIs, unless they are passed on as global variables.

The *channel* string is of the following kind :
obA ! scB ! mdC ! D

with A: number of the onboard channel through which the signal is multiplexed
B: number of the SCXI chassis
C: number of the module slot in which the module that is addressed can be found
D: channel to address

For example, the string ***ob0 ! sc 1! md1 ! 0*** means that channel 0 on the module in slot 1 of the SCXI chassis 1 , which is multiplexed through the onboard channel 0, is used.

Following this scheme, even more than one channel can be addressed at the same time, by typing the single channel numbers divided by commas, or a sequence of channels can be addressed by typing the first and the last number divided by a colon.

The device number corresponds to the slot into which the data acquisition board is plugged in the computer. In this case, the device number of the DAQ board was 7 .

IV. CHAMBER MODIFICATIONS

The changes which were to be accomplished in the TVAC operation made several mechanical changes necessary. Those changes included both mechanical design tasks and the installation of new electrical components in and around the TVAC .

A. ELECTRICAL COMPONENTS

First of all, the electrical wiring had to be changed and expanded when the SCXI hardware was installed. The installation of the six relays used to bypass the manual switches on the front panel of the TVAC shielding had to be done by connecting their contacts to the corresponding contacts of the switches.

On top of the chamber a connector box with plug-in sockets for 24 thermocouple connectors was installed, close to the inlet into the chamber. The box is connected to the SCXI-1300 terminal box mounted on the front of the SCXI chassis. The chassis is located at the bottom of the cart on which the TVAC is mounted. This connection was made by means of an isolated cable containing 24 wires, which was placed along the backside of the TVAC chassis .

Furthermore it was necessary to install a DC power supply next to the SCXI chassis from which a cable leads to the SCXI-1162 module in slot 4. The power supply is needed to deliver the power to actuate the 6 relays . A second cable coming out of the front of the SCXI-1162 module leads into the TVAC shielding through a hole on the back of the chamber. It connects the module with the relays and is used to change the state of the relays by sending electrical pulses.

The last wiring needed was a cable from the SCXI-1320 connector box on top of the SCXI-1120 module to the TVAC temperature controller which is located on the backside of the housing front door. This connection establishes the reading of the TVAC

temperature directly from the control display, which is delivered by a fixed thermocouple on the inside of the chamber.

All these changes had to be made according to the wiring diagrams of the chamber. By looking at the diagrams various facts about the TVAC control schematics were found out. In order to avoid a malfunction it was important to determine which switches correlate with each other to make sure that they are switched together in the computer-controlled mode.

B. MECHANICAL ITEMS

1. Relay Holding Device

The Relay holding device is a rather simple part, because its only function consists of attaching the 5 relays, which bypass the TVAC temperature functions, to the inner door of the TVAC housing . In order to allow visual control of the relay setting, it was decided to construct the holding device in a way that the relays can be looked at from their sides , if the TVAC housing door is open. This is of special importance when it comes to the start-up setting for the TVAC control loop program (see chapter V)

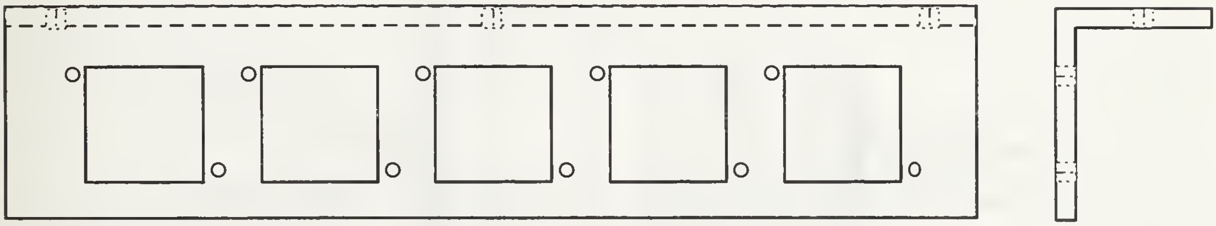


Fig. 11 : Shape of the Relay Holding Device

2. Connector box

a. Requirements

The main reason for replacing the existing cable-connector and constructing a new one was the idea to expand the usefulness of the chamber by allowing multiple functions at the same time. For future applications the connector box has to provide a connector for the thermocouple wires (a total of 18 cables) and at least one additional connector.

A second connector will be used for the spacecraft component or subsystem that is being tested inside the chamber, to provide power supply, take measurements and establish data links. Additionally, it was considered to attach four coax-cables to insert light into the chamber to simulate sunlight for applications like solar panel testing .

Another requirement which had to be taken into consideration was the ease of exchangeability of the whole box or at least the connectors : dependent on the application, the test operator should be able to use additional plugs or a completely different set of connectors without the need to design a completely new connector box.

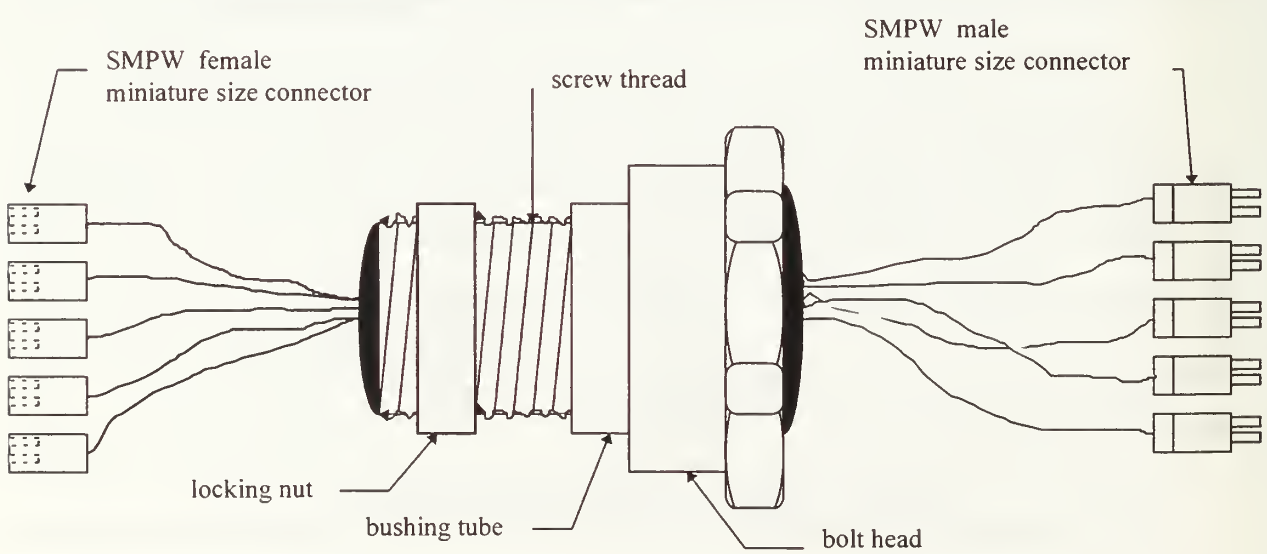


Fig. 12 : Thermocouple Connector

The connectors which have to fit into the box are those which were used before this thesis, but only one at a time. They have the shape shown in Fig. 12.

The most important criteria in the design phase, though, was to allow for the fact that the connector box is connected directly to the opening of the inner TVAC chamber. Thus, the part has to tolerate high vacuum of up to 7.5×10^{-8} torr and temperatures between +177 and -73 degrees Celsius, which had a decisive influence on the design. Therefore, the design criteria of the box can be summarized as follows:

1. The box has to be vacuum proof and temperature proof
2. It must to be durable
3. It has to hold two connectors and provide room for additional four coax-cables for future applications
4. It must be possible to open it in order to install connectors, cables, thermocouples etc.
5. The box must be modifiable to provide variability.
6. The box has to be removable from the TVAC to allow the installation of different types of connectors or a simple lid.
7. It has to fit on the already existing opening into the chamber.
8. The very sensitive ion-gauge, which is located close to the opening, has to be protected against impact.

b. The Part

Taking all this into consideration, it was agreed to machine the box out of a single block of aluminum with the following features :

- rectangular horizontal shape with round corners providing protection for the ion gauge and preventing injuries.

- thick walls of half an inch to provide stability and stiffness
- a removable lid on top, into which the connectors are to be mounted. This provides exchangeability of the lid and, therefor, the use of different lids with different connector settings.
- all connections between different parts have an o-ring seal to prevent the part from leaking. To make sure, the part had to undergo a Helium-Leak-Test after completion.

The upper part, which is the actual box, has a rectangular orthographic top view of 4 inches by 8 inches. Its corners are rounded with a radius of 0.312 inches on the inside and 0.812 inches on the outside. The box is 4.0 inches high and provides a wall thickness of half an inch . The foot of the part, which connects the connector box to the TVAC opening, is a round plate of 3.575 inches diameter. This junction between the TVAC housing and the connector box is sealed by an o-ring, positioned in a groove around the opening. Using the existing o-ring of the TVAC had the advantage that the air-tightness of this connection had already been approved.

The plate is connected to the upper section by a tube of 1.250 inches length which serves as the feedthrough for the cables from the connectors in the lid. Due to the number of at least 25 cables which are supposed to feed through this tube, it had to be given the biggest inner diameter possible, which in this case was the diameter of the inlet into the chamber. Thus the inner diameter of the tube is 2 inches.

The whole box is mounted onto the opening in the housing by means of four $\frac{1}{4}$ - 20 screws (0.265 inches) , using the holes which already existed from the former connectors .

The lid was made out of an aluminum plate of 1 inch thickness. In order to provide air-tightness and stability, it was decided to give it a border of 1/4 inch thickness into which the body of the box fits very closely. This allowed the use of larger screws which helps to create more compression.

The lid contains two simple through-holes of 1.635 inches diameter for the connectors. The four coax-plugs have not been prepared yet, but room for them was left in the middle of the lid. The attachment of the connector plugs into the lid is made by simply plugging them into the mounting holes. They are then tightly attached to it by means of locking nuts which are screwed from the bottom upwards.

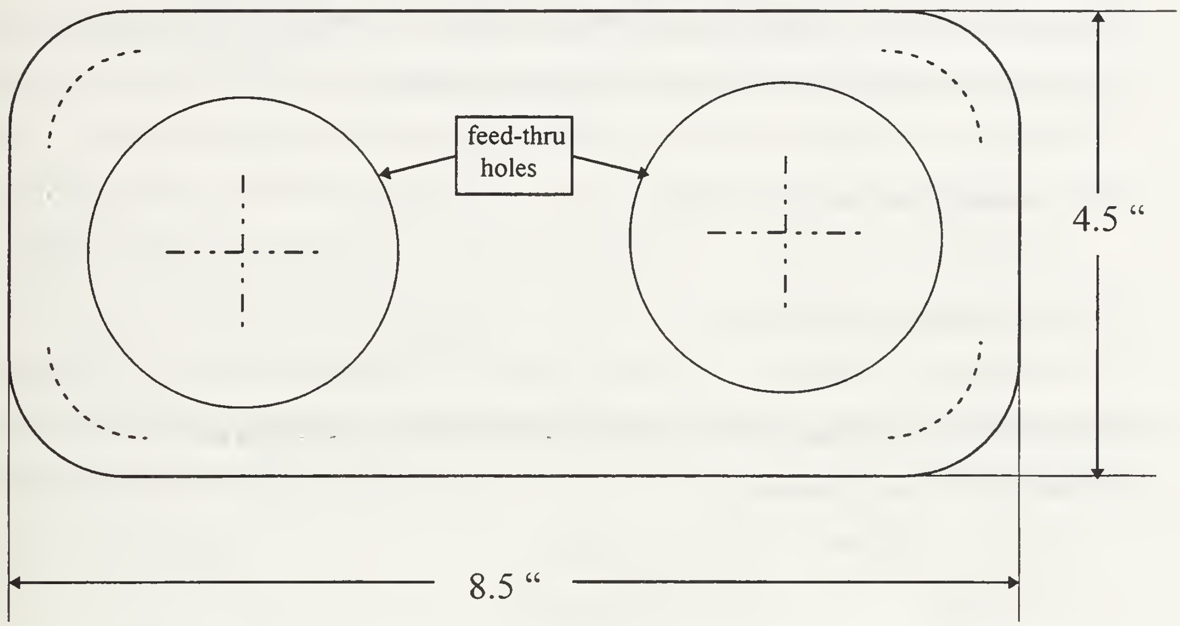


Fig. 13 : Sketch of the lid

The connections between the lid and the connectors are sealed hermetically by o-rings. This creates no new o-ring groove in the lid, which could have been a possible source for leaking.

The connection between lid and box was a critical item in the design, because the upper surface of the box, into which the groove for the o-ring-seal was to be inserted, has a rectangular shape. Normally, though, o-rings are only delivered in a round shape, and the determination of the appropriate o-ring is also done by the inner diameter and the width of a <u>round</u> o-ring groove. As the size of the o-ring groove was not yet clear, the decision had to be made the other way round.

By taking the size of the rectangular surface and translating it to the dimensions of a circle, the possible o-ring size was determined.

possible length = Σ side lengths

= 20.09136 "

= perimeter of a circle

= Π * D => $\underline{\underline{D = 6.399 "}}$ = possible diameter of o-ring

Then, by looking on the corresponding page and column in the Parker o-ring catalog , the o-ring type was determined according to the following criteria :

- expected heat profile of the part
- high vacuum resistance
- non-military application

According to this, 12 o-rings of the type ID 2-165 V747-75 were ordered .

The o-ring has the specifications :

ID (inner diameter) : 6.48 ± 0.04 inches,

W (width) : 0.103 ±0.003 inches

The V747-75 o-ring is made of Viton, a Fluorocarbon Elastomer, which can withstand temperatures between + 300 °C and - 250 °C.

The size of the o-ring groove was determined to be :

Width : 0.120 ± 0.003

Inner diameter : 0.077 ± 0.003

These values had to be translated into a rectangular shape with round corners again , which resulted in the final dimensions presented in the drawing .

c. Helium Leak Test

After receiving the o-rings, the whole part was subjected to a Helium-Leak-Test. This kind of testing is used to determine if a part which is designed to undergo high pressure differences between the inside and the outside can actually provide a seal.

Using an ***Alcatel ASM 121 h Helium Leak Tester***, a vacuum was created on the inside of the connector box. After a certain amount of time, Helium was sprayed onto the outside of the part and especially close to those locations, where o-rings are used. The box underwent the test twice: First it was tested with the lid on top, before the through-holes for the connectors were cut out. This way it could be determined if the o-ring seal between the lid and the box worked properly.

After the mounting holes had been cut out, the connectors were attached to the lid and the test was repeated. Thus, if any leaking had been detected throughout the second run, it would have been caused by the connector-to-lid seal. Since both tests were successful and no leaking was detected at all, the connector box was finally mounted onto the TVAC housing.

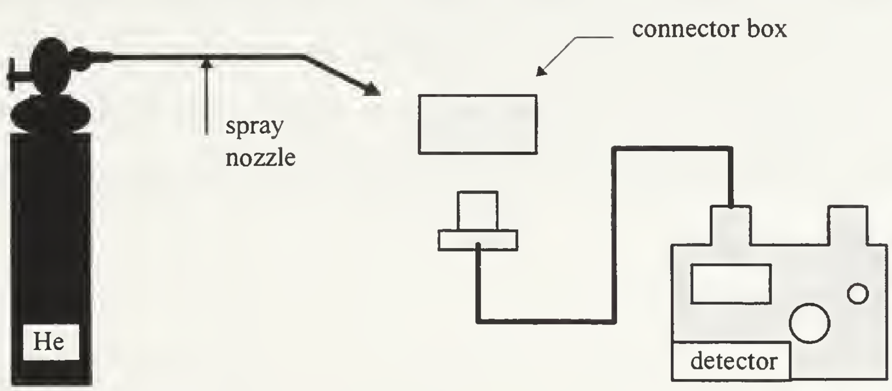


Fig. 14 : *Schematics of Helium Leak Test*

3. The CAD Program

The complexity of the connector box on top of the inlet into the TVAC chamber suggested to design the part by computer, using the IDEAS™ Master Series 2.0 CAE/CAD/CAM software by SDRC.

IDEAS is the abbreviation for **I**ntegrated **D**esign **E**ngineering **A**nalysis **S**oftware and is an integrated package of mechanical engineering tools. It is composed of a number of application software modules, each subdivided into several tasks, which can all be executed from a common menu and share the same database. IDEAS is the CAD program which is most commonly used by the SSAG, because it provides a great variety of features and possibilities.

Besides the simple design of mechanical components, the program allows finite element analysis, functional simulation and more. Due to the program's complexity, a time span of about two weeks was needed to learn the basic functions of IDEAS®, following a tutorial provided by SDRC, before the actual design task could be started.

V. SOFTWARE DESIGN

A. BASICS

The LabVIEW software package contains a great number of example libraries for all kind of applications. As the SCXI hardware and LabVIEW are both trademarks of National Instruments, a special library with Virtual Instruments (VI) for SCXI and Data Acquisition is also included.

It was therefore decided, to look if some of the VIs of the package could be used, perhaps with some changes and modifications. It appeared that the packet included VIs to read thermocouples, to read bit patterns from, and write them to the SCXI hardware and to read voltages.

These VIs fulfilled the basic ideas of the application to be developed, but they did not take into account the fact that the DAQ board in this case was only capable of handling one data buffer at a time. Thus, problems occurred when several VIs , running at the same time, created buffer overflows, forcing LabVIEW to display error messages and to stop, and sometimes crashing the computer system. Since it was important to read the thermocouples and the voltage of the TempTenn controller simultaneously, as well as to send bit patterns to control the TVAC switches, another way had to be invented.

An additional design topic was to create a user friendly graphical user interface (GUI) from which the TVAC could be controlled completely, and comfortably. It became clear that this task could only be fulfilled after the whole program was designed and tested.

B. THE VIRTUAL INSTRUMENTS

The whole software design project can be divided into 6 parts :

1. Data acquisition
2. TVAC control
2a. TVAC test I
3. Control loop
3a. TVAC test II
4. Graphical User Interface (GUI)

The tests will be described in detail in chapter VI. " TVAC tests "

1. Data Acquisition

The first phase in the software design was to use LabVIEW and the SCXI hardware for data acquisition. The data acquisition concept within LabVIEW is different for each VI, which makes it extremely difficult to have several VIs run at the same time, if they differ in the acquisition concept that is used.

LabVIEW provides three different data acquisition facilities :

I. Immediate nonbuffered I/O

Using immediate I/O operation, a single scan is read from or a single update written to a given group of channels. In this case, no scan clock or update clock is used . Hardware-timed interchannel delay can be provided by the channel clock.

II. Timed nonbuffered I/O

Nonbuffered I/O operations cause LabVIEW to sample the input or update the output. This means that the data acquisition is synchronized with a clock, to be more precise, onboard timers/ counters are used. Thus, the on-board data acquisition board acquires or generates data at precise intervals. This type of data handling is mostly used when a precise timing is needed, but a large buffer is not desirable. A loop, for example, does not have to be timed by a VI but is timed precisely using hardware.

III. Timed buffered I/O

Whenever accurate hardware timing is needed for an I/O operation, timed buffered I/O operations have to be used. With this type of I/O, the data is transferred to, or taken from a buffer. The transfer takes place at timed intervals, for which clock and trigger conditions have to be specified beforehand.

A distinction is made between simple buffered I/O and circular buffered I/O. Whereas simple buffered I/O writes the data into the buffer just once, circular buffering fills the buffer more than once, circularly.

The AT-MIO 16E-10 board can automatically acquire a **timed sequence** of samples on one or more channels, using a method called interval scanning. The scan rate is controlled by the scan clock, and is normally programmed in Hertz. The board is programmed to either acquire a fixed number of scans or to acquire continuously. When the board acquires more than one channel, onboard memory stores a list of the channel numbers and the associated **input limit settings**. The input limit settings are given in units associated with the channel they describe and determine the range which is scanned by the computer. The list contains up to 16 entries, and at each scan interval all the channels are sampled. The rate of sampling is controlled by the channel clock. The channel clock can be controlled by the Clock Config VI for example, which can be found in the *Functions* drop-down menu under *DAQ -> Advanced.*

The LabVIEW Example VI Library contains various VIs which acquire analog inputs (referred to as AI in the following). Although the signal acquired is always a voltage, however, it can represent temperature. Since the DAQ board is not capable of reading from two different SCXI modules at the same time, the idea of using two different VIs, one for temperature and one for voltages, was not possible.

As each of these VIs uses a different clock controller or different settings, a buffer conflict occurred every time when two VIs were run at the same time. LabVIEW displayed an error message, and sometimes even the computer crashed.

Although it would have been desirable to have two separate VIs for the two tasks in order to display them on the screen separately when needed, another solution was sought.

The SCXI Temperature Scan VI, which is part of the *Functions -> DAQ-> DAQ Utilities* library, is a VI that reads thermocouples and voltages from a number of channels.

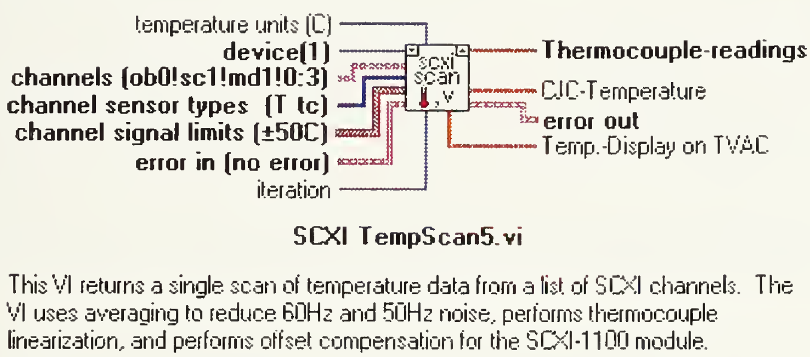


***Fig. 15** : Help-Display for SCXI Temperature Scan VI*

The SCXI Temperature Scan VI reads analog input data, returning a **single scan** from a list of SCXI channels. For each channel or range of channels in the channel string array it can be determined if the channel sensor type is a thermocouple or a voltage channel. Data from thermocouples is converted to temperature, data from voltage channels is scaled to volts. For each channel, the signal limits can be specified in units of the desired output.

The VI averages the data and therefore reduces noise in the 50 to 60 HZ range. Furthermore, thermocouple linearization and offset compensation are performed. The data, which is read one scan at a time, is returned in a one dimensional array, with each input in the corresponding unit.

The iteration terminal VI determines if the VI initializes on each call. Initialization in this case means that the VI performs offset and cold junction readings and configures the analog input. Initialization is only performed when “0” is entered, any other value has no effect.

The SCXI Temperature Scan VI is a so-called **advanced VI**. This means that the VI uses other VIs as subVIs on different sub-levels and is already “ready-to-use” for a certain task. The user only has to connect the wires to it and to obtain an output.

This VI uses several subVIs, the most important of which are listed below:

- AI Control
- AI Read
- AI Config
- Clock Config

(Their descriptions can be read in the *Data Acquisition VI Reference Manual.*)

To solve the problem with the buffer conflict, the SCXI Temperature Scan VI was modified several times.

Scanning the TVAC display voltage from the SCXI 1320 module, which is done through the channel *ob0! sc1! md2! 0*, was included. The VI was changed in a way such that, using the *Acquire and Average* VI from the library, the voltage data is now read synchronously with the thermocouple data within one scan. The data is then appended to the array of the thermocouple temperatures.

The TVAC display is therefore read as many times as the thermocouples and the data can use the same buffer at the same time. The TVAC display was furthermore given a single output connector.

In order to check the cold junction temperature (CJT), other changes had to be made. The SCXI Temperature Scan VI automatically reads the CJT and calculates cold junction compensation, but without displaying the value which is used for the calculation. Thus, similar to the TVAC display data, an *Acquire and Average* VI was corporated into the diagram, to read and then display the CJT. Like before, another output connector was added to display the CJT on the VI. (The figure above shows the final version of the VI which was called SCXI TempScan5).

After the changes in the VI had been made, the SCXI-1120 module and the SCXI-1320 terminal block had to be software-configured using the WDAQConf file in the LabVIEW directory. According to instructions given by a National Instruments representative, the gain was set to 1.

To display the TVAC temperature, the voltage data, taken from the TempTenn , has to be converted into temperature. The output range of the controller is -2.5V to +5V. As a temperature range from -99.9°C to 199.9 °C can be displayed, the following equation was determined:

- 99.9 °C = -2.5 V

199.9 °C = 5.0V

$$\Rightarrow T = -99.9 + \left(\frac{199.9 - (-99.9)}{5.0 - (-2.5)} \right) \bullet \big(V - (-2.5)\big)$$

This means a voltage-to-temperature ratio of $\approx$ 0.025 Volts / °C .

The zero °C is calibrated to 0 VDC, 24°C $\approx$ 0.6 VDC.

Since the TVAC display had to be reevaluated and both TVAC display and CJT had to be modified from an array type to a value output, a new VI *called reading thermocouples2.vi* was created.

Using the SCXI TempScan5.vi, this VI reads the TVAC display, the cold junction temperature and a predetermined number of thermocouples.

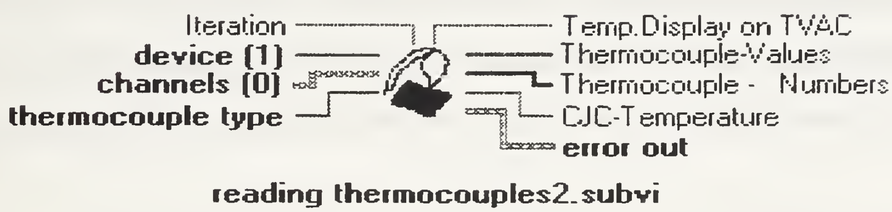


Fig. 16 : Help window of reading thermocouples2.vi

2. TVAC Control Program

The second step was to create a program, that was capable of controlling the TVAC. The idea was that, although data acquisition of the thermocouples, the CJT and the TVAC is done continuously, the operator can control the relays in the TVAC by clicking switches on the screen. The curve is displayed on the VI so that any change in the curve, caused by a different switch setting, can be seen on a chart. Furthermore, the saving of data to disk had to be designed in order to allow testing, which was the next step.

This task was divided into three parts :

a. Create a program to switch the relays

b. Integrate the data acquisition into the control program

c. Program a "save to disk" feature.

a. Switching the Relays

The relays are switched by electrical pulses, which are given from the SCXI 1161 module. As the SCXI-1161 is a digitally operating module, a VI for digital output was needed. LabVIEW libraries provide an example VI called "Getting started digital I/O ", (I/O stands for input and output) which can be found under *Examples -> DAQ -> run-me.llb* .

Getting started digital I/O VI writes a bit pattern, represented by boolean switches, to a set of digital lines. The pattern, which is displayed as a LED pattern , is converted to a decimal number and then passed to a subVI *Write to digital Port VI.* As this example VI fulfilled all the basic needs of the control program, it was decided to modify it in such a way that it could be used for the application to be designed.

The whole program was put into a while loop, and the LED pattern was replaced by a pattern of six switch controls. Each of the switches represents one of the relays, and clicking them toggles the relays.

b. Data Acquisition

In the next step, the *reading thermocouples2 .vi* was added into the *WHILE*-loop to allow data acquisition. Most of the input parameters, which were to remain constant throughout the run, were passed on as constants.

Moreover, two additional displays were added to the VI for convenience :

1. A chart, to display the thermocouple temperatures and the TVAC temperature as waveforms. In an earlier version of the VI, the chart was made a subVI called *display chart.vi* which would only appear when a button was pressed.

However, throughout the tests it became clear, that it was important to display the temperatures permanently in order to have a quantitative idea of the behavior of the curve.

2. Two timer displays, one showing the start time of the VI and one to display the present time. These times were included into the SAVE option, to allow a real-time display of the datapoints under Microsoft Excel. Furthermore they proved to be convenient for the operator to have the timers in view on the screen.

c. Save to disk

Since test results have to be evaluated and interpreted, data storage to disk was probably the most important feature of the control algorithm.

In order to save the data and record the switch setting in a way which is easy to read, usable by Microsoft (MS) Excel, two new VIs were created:

- *String setup .vi*

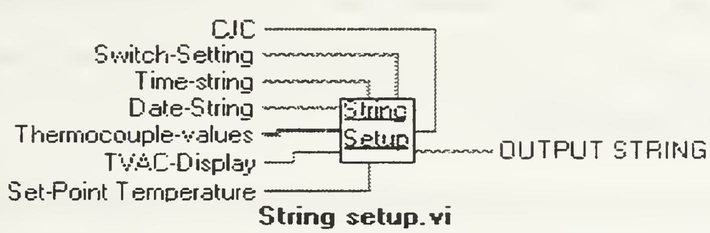


Fig. 17 : Help window for String setup.vi

This VI collects all the data to be saved and converts it into a single string which will represent one row in a spreadsheet. First of all, the date and time when the data is taken, is saved. Appended to this are the TVAC temperature, the CJT and the

setpoint temperature. followed by the switch settings. Then, the thermocouple values are concatenated.

As the data types are different. several functions are used in this VI to convert the numbers, arrays and strings into a format that can be placed in an Excel spreadsheet file .

- *Switch setting . vi*

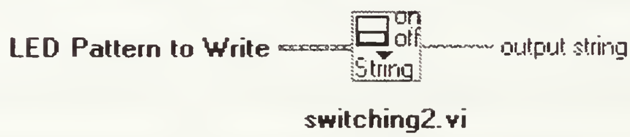


***Fig. 18** : Help-window for switching2.vi*

This VI is used to convert the switch setting of the relays, which is a bit pattern, into numbers. As the goal was to display the switch setting in the same Excel chart as the temperatures, it was decided to display each switch by an individual line.

The bit pattern is converted into a boolean number 0 (off) and 1(on) and then multiplied by a certain integer, which is characteristic for each switch. The numbers are 100 (cold trap), 95 (subzero), 90 (ambient cooling), 85 (heat), 80 (doorheat) and 75 (Bypass relay, simply called relay #6).

The best method to display the resulting chart is by using a spreadsheet under Microsoft Excel. See Appendix C for a brief description of the procedure necessary to import the data into Excel.

As LabVIEW does not provide timers in the sense of counting seconds, a count loop was created, using the Function *Get date / time in seconds* twice. These timers display the amount of seconds that have elapsed since 12:00AM, Friday, January 1, 1904GMT.

One of the counters is started when the program begins. Another one is located within the while loop. By comparing these two counts, a time difference in seconds is created. The program allows the input of a certain amount of time in seconds, after which the data is saved and the timer loop is reset to zero.

Additionally, the switch setting is passed on through shift registers which allows a comparison of the last value with the present one. Every time a change is detected, which means that a switch has been clicked, the save algorithm is also executed.

These features mentioned above where put together into a VI called *TVAC testing7.vi* .

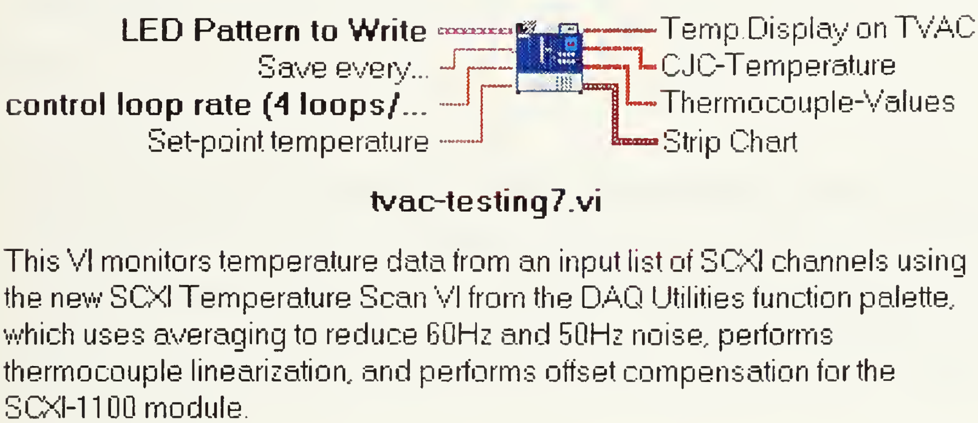


***Fig. 19** : Help window for tvac-testing7.vi*

3. TVAC Control Loop

The third step of the program development was to expand the program further: Instead of having an operator switch the TVAC controls “manually” by computer, the computer should be able to reach and maintain a ***setpoint*** by itself.

The setpoint is an input from the operator to set the desired temperature. The

program then tries to reach the setpoint as quickly as possible. The setpoint is determined by a pop-up window which appears when the program is started. While the program is running, a boolean switch has to be pressed to have the window come up and change the setpoint.

The TempTenn controller allowed an overshoot over the setpoint of more than 15 degrees, depending on the setpoint temperature. Furthermore the setpoint was maintained within a temperature range of at least ± 5 degrees, which was not very precise.

Another task was to create an algorithm capable of approaching a setpoint smoothly and maintaining a temperature within a narrow range. This was achieved by the following :

After multiple TVAC tests, the following characteristics of the TVAC were found :

- After switching off the heat, the transfer of energy in form of heat continues for approximately 3 minutes. This effect takes place, because the heating device in the chamber takes a long time to cool off. The effect lasts even longer if the chamber has remained at a high temperature before. This effect will be referred to as " overshoot ".

- Any change in the switch setting, be it heating or cooling, takes a delay of approximately one minute to take effect, which means that the shape of the curve within that delay time is rather unpredictable.

- The higher the setpoint, the longer are the periods of heat, which are necessary to maintain at setpoint level. In order to avoid a large overshoot, the heat has to be added in sequences of heat pulses alternating with wait periods (periods of no heating).

- The ratio of periods in which the heat is on and off influences the climb rate of the temperature. The longer the heating period, in comparison to a period of no heating, the steeper the temperature rises. But, although the ratio is important, the time of an uninterrupted heating interval is even more important.

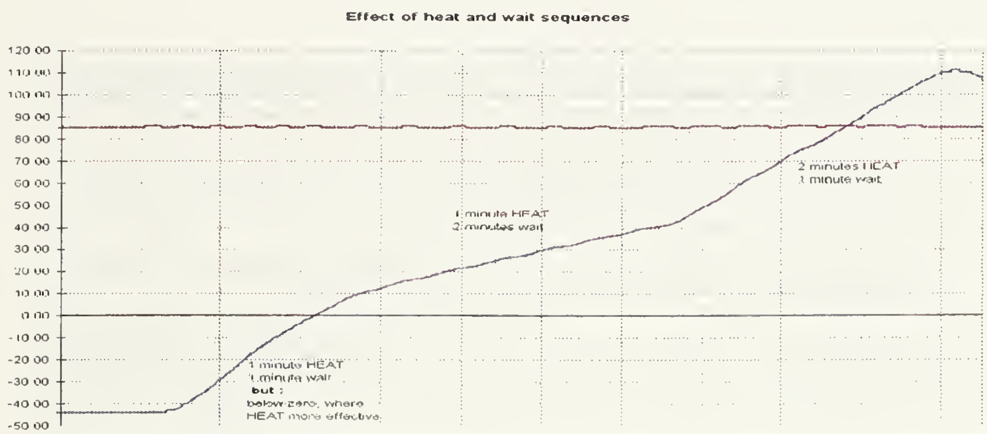


Fig. 22 : Effect of heat and wait sequences

- As expected, the temperature curve is not a linear function:

  When cooling is performed, be it natural, *AMBIENT COOLING* or *SUBZERO COOLING*, the curve decreases with an exponential decay. The rate of decrease is dependent on the temperature, the type of cooling used (passive, ambient cooling, subzero) and the temperature difference to overcome .

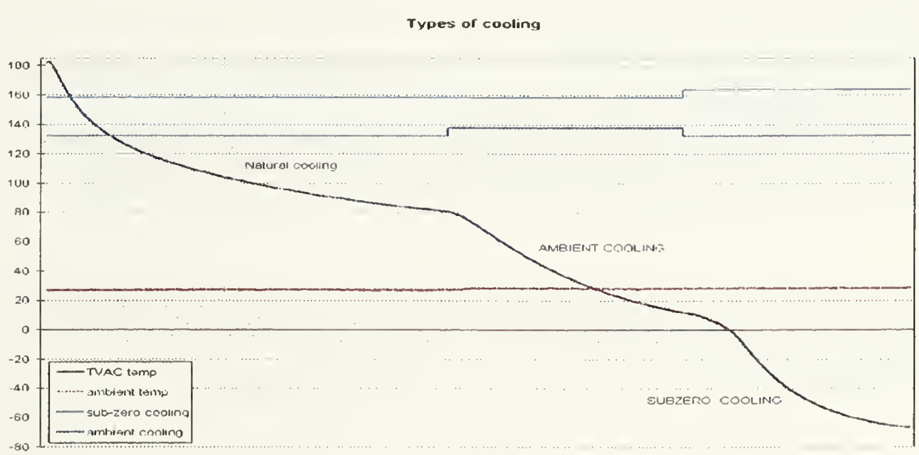


***Fig. 20** : Different types of cooling*

When heating is performed, the curve increases rapidly, but the slope decreases slightly as a function of time. Within the temperature range from 80°C to 160°C, the temperature curve comes very close to a linear function when the *HEAT* has been on for a while.

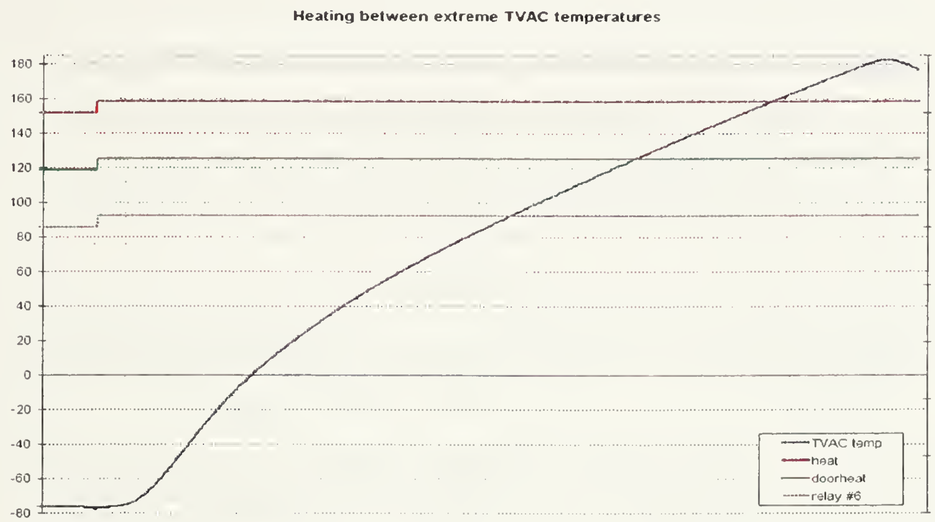


***Fig. 21** : Heating between extreme TVAC temperatures*

In addition to this, the following characteristics and restrictions were learned from the TVAC manual :

1. The cooling works as a differential function, using a refrigeration system. The refrigeration system consists of two compressors and a condenser fan. The cooling efficiency is dependent on the TVAC temperature and the ambient temperature.
   - If *AMBIENT COOLING* is used, cold gas is directed through the cooling coils.
   - *SUBZERO COOLING*, which is faster and can reach the lowest temperatures, uses liquid refrigerant for higher cooling capacity. Thus, the rate of decrease it much higher for *SUBZERO COOLING.*

2. Although it would be allowed to use *SUBZERO* at high chamber temperatures, it was agreed to use it only below 40°C. Doing so is easier on the refrigeration system and cooling is more efficient .

3. Switching on and off any of the cooling functions is not desirable because it would stress repeatedly mechanical parts. According to this, the maintain functions below ambient temperature have to be done by switching *HEAT* on and off and leaving the cooling switches in the closed position .

4. The *HEAT LOAD VALVE* works like a low power heater. It was used by the TempTenn controller, together with the *AMBIENT COOLING*, to perform slow cooling. After bridging the TempTenn controller, the valve is operated continuously as long as relay #6 is closed (ON). As it would be a waste of energy and the cooling slope would be less steep, it is not desirable to cool with relay #6 on.

5. When the HEAT and the DOORHEAT functions are run simultaneously, switching off HEAT must also result in switching off DOORHEAT. This is the way the TempTenn controller handles it.

6. The doortrap, whose function is to reduce backstreaming of oil from the diffusion pump to the chamber, may only be operated when the chamber has been roughed down to about 100 microns. If the trap is chilled at higher pressures, it may get contaminated.

7. As a matter of safety, a manual switch on the front panel of the TVAC has to be added in order to allow the operator to take over the TVAC control from the computer.

Using the basic functions of *TVAC testing7.vi*, a control loop program was designed which should be able to control the TVAC at any temperature, no matter what temperature function was chosen, according to these operational criteria.

The complexity of the TVAC can be shown by taking all parameters into the following equation for the temperature curve :

$$\frac{dT}{dt} = T' = f(T\,;\,t\,;\,T_{amb}\,;\,S\,;\,p\,;\,\Delta e) \qquad (i)$$

with : T' = change of the TVAC temperature

T = TVAC temperature

t = time

$T_{amb.}$ = ambient temperature (room temperature)

S = switch setting = $f(S_{HEAT}\,;\,S_{COOL}\,;\,\text{RELAY \#6})$

with S_{HEAT} = switch setting used to heat

S_{COOL} = switch setting used to cool

p = pressure in the chamber

Δe = energy production by test device in the chamber (not present in this case)

where the TVAC temperature **T** is a function of the following :

$$\mathbf{T} = f(T_{START}\,;\,t_{HEAT\_ON}\,;\,t_{DELAY}\,;\,t_{OVERSHOOT}\,;\,S_{HEAT}\,;\,T_{MAX\_before}\,;\,t\,(T_{MAX\ before})) \qquad (ii)$$

with : T_{START} = start temperature when *HEAT* was turned on

t_{HEAT_ON} = time that *HEAT* was turned on

t_{DELAY} = delay time until changes take effect

$t_{OVERSHOOT}$ = time after switching *HEAT* off, while heat is still being emitted

S_{HEAT} = switch setting that is/was used to heat (*HEAT* and/or *DOORHEAT*)

T_{MAX_before} = highest temperature the TVAC was at within the last few hours

$t\,(T_{MAX\ before})$ = time that highest temperature was held

The first idea was to transform the TVAC characteristics into the s-jw plane , using Laplace-transformation. By means of a root locus curve, taking into account the parameters, the future behavior of the slope could possibly have been predicted. Since the number of unknown parameters was much too large and a function for the TVAC behavior could not be found from the test results, it was decided that the control problem could not be solved this way.

The second effort made was by estimating a simple time constant in order to predict a future temperature. The time range in which the temperature would overshoot was discovered to be always between 3 and 4 minutes. It was attempted to estimate the point in time at which the curve would reach the setpoint. The amount of heat could be adjusted depending on whether the setpoint is reached earlier or later than estimated. Since the prediction cannot be made due to the large number of parameters, this method was also abandoned.

It was then decided to create the loop program as a sequence of many *CASE* structures, which could all be adjusted to the temperature according to the test results. The description of these structures can be found in the following chapter.

Some major parts of the program will be described briefly. As the heating and the cooling functions are completely different functions, they were treated as two different cases. The whole *WHILE* loop was put into a big sequence structure to allow warning messages to come up during the very beginning. (see: " User Interface").
The subVIs which had to be programmed for the control algorithm, which was called *control loop.vi,* are now described .

As it appears from examining the equations (i) and (ii), the amount of heat that has to be admitted to the TVAC depends on various parameters. If any overshoot is to be avoided , at least the current temperature level and the rate of increase or decrease of the curve have to be taken into account. This is a rather complicated problem, as even the original TempTenn controller worked by allowing great overshoot. Temperatures were maintained by oscillating within a temperature range of ± 5 °C.

Learning from the experiences of operating the chamber using the TempTenn controller, the following assumptions for the *control loop.vi* were made :

a) To avoid overshoot, the *HEAT* function may not be operated continuously within a relatively wide range of the setpoint.

b) The slope has to be adjusted to the difference in temperature from the setpoint by using timed loops consisting of heat and no heat sequences

c) Maintaining a given temperature is only done by switching the heat functions. If the setpoint is below room temperature, the cooling functions remain operating.

d) The setpoint is to be reached as precisely as possible, with no more than $\pm$ 2.5 degrees oscillation.

a. Avoid Overshoot

In order to avoid overshoot, the *HEAT* function may not be operated for too long at a time and it must be switched off long before reaching the setpoint. As it was determined during the tests, the TempTenn controller allowed overshoot of more than 15 degrees, depending on the setpoint. Furthermore it became clear that the overshoot is a function of the time the heat was admitted, the temperature level the TVAC and the current slope of the heat curve.

$$\text{overshoot} = f\,(\, t_{HEAT}\, ;\, T_{TVAC}\, ;\, T'\,)$$

The tests gave these results:

- The highest overshoot that could be achieved throughout testing was 28 °C
- The longest time the curve would overshoot was approximately 3 1/2 minutes
- The maximum climb rate is 11.43 °C per second

Furthermore it was found that the overshoot was higher if the *HEAT* function was started while the temperature curve was falling. The amount of heat delivered in that case was much higher, because the negative slope had to be stopped first, before an increase of the temperature could start. Thus heat was admitted for an additional time of more than two minutes which resulted in a longer lasting overshoot.

According to these results, it was decided to control the point at which the *HEAT* function changes from continuous operation to periodical operation by the following algorithm, which was called *Determine full heating time.vi :*

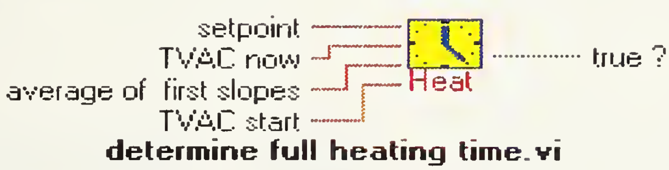


Fig. 22 : *Help Window for determine full heating time.vi*

1. If the difference between the start temperature and the setpoint exceeds 50 °C, the HEAT will change from full heating to periodic operation at a temperature difference of 30 °C.

2. If the initial difference is less than 50°C, the mode of HEAT operating will change as soon as 40 % of the difference is overcome.
   (or 30 % if the slope was negativ at the beginning of the loop)

3. If the initial difference is even smaller than 5°C, the HEAT will only operate periodically, never giving full heat.

With this algorithm, overshoot of more than 2 degrees never occurs. In case the curve does not increase fast enough, the periodically given heat prevents the slope from becoming too low: By adjusting the amount of heat to the slope, a different loop with longer heating periods is performed.

b. Adjustment of the Slope :

Trying to adjust the amount of heat to the current slope, it was necessary to design a VI which would pass on a certain number of consecutive TVAC temperatures. Only by comparing multiple values, a future behavior can be predicted, though still very imprecisely. The VI which was created for this purpose is called *array.vi :*

Fig. 23 : Help-window for array6.vi

The VI displayed above does not represent the latest version of *array.vi* because several changes had to be made until an accurate method for predicting the curve was found. However, all later versions are modifications of the one shown above.

The VI takes the current TVAC temperature value and appends it to an array of size *dimension of array* . The oldest value within the array is deleted by shifting the line index once. The new array is passed on to the next iteration by means of shift registers.

As in Fig. 23 above, the output connectors available are the average of the values, the second to last value and the present and past temperature change. Past temperature change means the slope 10 seconds before the present reading.

Attempting to adjust the amount of heat to the slope, different trials were made.

The first attempt was to take a change in the slope as a condition under which the heat input has to be adjusted. The problem with this was to find a means for determining if the slope changes.

Since taking the slope between two points on the temperature curve would not be an accurate measure, as the program needs about 1 second for of one loop iteration, a sequence of 3 different versions of *array.vi* was built.

1) A set of the last 15 TVAC temperatures is carried on through ***array8a.vi***. Every time, a new value is put in, the climb rate between the first and the last point in this array is calculated by $\Delta T / \Delta t$. The climb rate is then forwarded to the second version, *array8b.vi*.

2) *The **array8b.vi** VI* stores the last 10 climb rates which were calculated from *array8a.vi*. Everytime a new entry is made, the average climb rate is determined, using the *average.vi* function from the *functions -> analysis -> statistics* library. This average is passed on to the next array, **array8c**.

3) ***array8b.vi*** keeps a number of the last 10 rate averages. Every time a new input is made, the first and the last entry in the array are compared. If the oldest value is greater or equal to the latest entry, a boolean "true" is displayed, otherwise the comparison delivers "false".

Every time the climb rate decreases, according to that sequence of comparisons, a timed heating algorithm is executed. This algorithm consists of heat and wait intervals.

- a heat period, which is dependent on the difference in temperature between the current TVAC temperature and the setpoint.
- a wait period , which is dependent on the setpoint temperature

If the setpoint is below ambient temperature, the cooling functions are operated at the same time.

It became clear throughout the test, that this way of describing the slope did not work properly:

Since the program needs about one second until a new scan is acquired, the curve is actually not continuous, but consists of discrete values. The curve is built from a sequence of small lines, which can be seen on the chart in Fig. 24.

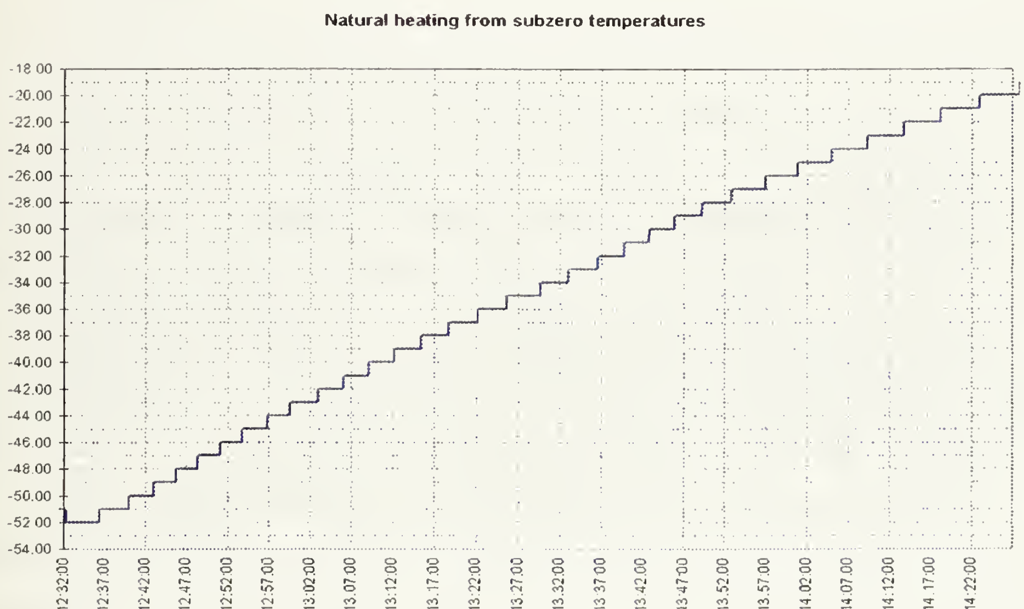


***Fig. 24** : Natural heating from subzero temperatures*

Some line pieces have a slope, others are straight lines with a climb rate of zero. So the calculated array of slopes consist of relatively many zeros. It is then possible that, although the slope is actually decreasing, the condition would become false for a short moment because one new average can be smaller than an older one. The problem that occurs then

is that the heat loop stops immedialtely, since it is performed only while the slope condition is true. Since the small heat algorithms are built in a way that the TVAC is to perform a short time of heating, followed by a wait time, a break in the heat loop means that heating is carried out, but the wait time is skipped. As soon as the condition of the decreasing slope becomes true again, the heat loop will start again without the assigned wait period. As a result, the amount of heat admitted to the chamber is too large, and an overshoot occurs.

Any attempts to change the condition for the change of slope failed, due to

- taking more data points results in too long a time to compare two slopes
- making more sophisticated calculations causes an even slower performance of the program
- allowing a certain range in which the values may differ does not guarantee an accurate comparisons of the change in slope anymore

It was then tried to adjust the amount of heat to the difference in temperature between the current TVAC value and the setpoint.

The second attempt tried to solve the problem from a different point of view:

Making the *HEAT ON* periods in the heat loop as long as the difference in degrees, and adjusting the wait times to the setpoint level, a new series of test was performed.

As the wait times between the heat intervals were properly met this time, the TVAC temperature approached the setpoint fast and continuously. The main problem with this method lies in determining the wait periods because they are not a direct function of the setpoint. After several tests were performed and evaluated, different case structures for different temperature levels were established .

Finally, the tests were all performed within acceptable ranges.

c. "Maintain" Algorithms

Maintaining the temperature is the most difficult task within the control algorithm. Due to the one minute delay between a temperature control command and its execution, no immediate feedback can be obtained. The effect cannot be foreseen at all, and long heating intervals can easily result in an enormous overshoot.

Thus the only way to prevent overshoot is to execute longer algorithms which consist of a series of several consecutive heat and wait intervals. These algorithms start as soon as a small temperature range around the setpoint is reached and the curve has a predetermined slope. As the length of the intervals depends on the setpoint temperature which is to be maintained, those loops differ from temperature to temperature.

As an example, the maintain algorithm for any setpoint exceeding 80 degrees is shown and explained .

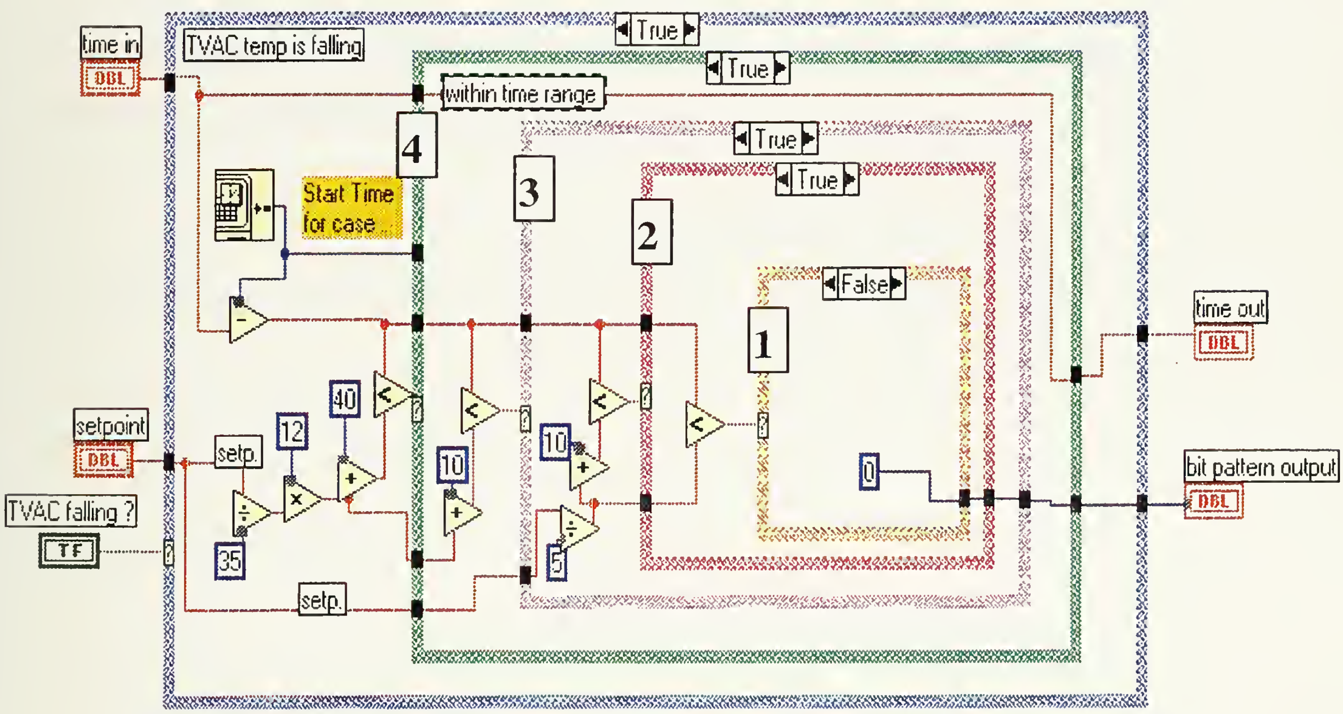


Fig. 25 : Maintain loop for temperatures > 80°C

Two timers, labeled "*time in*" and "start time for case" of the type *date / time in seconds* are used for this case. The timer *start time for case* starts within the outer *CASE* structure when the condition for the case is true. This timer is compared with the second timer (which is located outside of a large *WHILE* loop which is not displayed in this diagram and whose values are passed on steadily by means of shift registers.),

Both timers show the same time at the beginning. Whereas the timer within the *CASE* structure begins counting, the outer timer remains constant to serve as a start time reference. As long, as the difference between the two timers has not reached a certain amount of time (which is a function of the setpoint), a *CASE* structure is operated. The program checks the time condition from the inner most case (case1). Every time the condition for the case becomes false, the next outer structure is executed. Once the condition for the case titled "*within time range*" becomes false, the maintain loop expires. And, the time of expiration is passed to the outside WHILE loop.

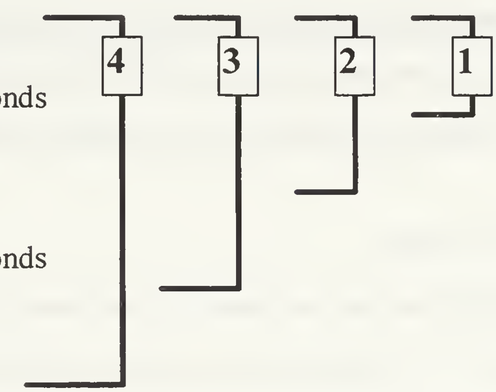
The time condition for which the cases are executed is shown in the following diagram (the numbers on the brackets display the cases) :

I. HEAT ON for $\frac{SETPOINT}{5}$ seconds

II. HEAT OFF for **10** seconds

III. HEAT ON for $\frac{SETPOINT}{7}$ seconds

IV. HEAT OFF for **30** seconds



By executing this algorithm only when the temperature is falling and below the setpoint it is made sure that there are even longer breaks within the loops than the 30 seconds wait time at the end. Usually the temperature starts increasing again after a single execution.

As the “maintain” loop is only supposed to start when a small overshoot has occurred and the temperature is falling, two small algorithms

a) overshoot

b) temperature falling

had to be generated to determine these conditions.

a) *overshoot :*

When the program is started or a new setpoint is input, the current TVAC temperature is saved. By comparing the setpoint and the start value of the TVAC, it can be determined if cooling or heating has to be performed. Furthermore, an integer 0 is sent to the shift registers.

On each iteration of the control program, it is checked if the TVAC value has already overshot. An overshoot happens, when cooling is to be performed and the TVAC value has reached any temperature below the setpoint. According to this, any temperature higher than the setpoint demands an overshoot in heating. When an overshoot is present, the integer which is passed on to the shift registers is given the value 1.

The case structure reads the value of the integer, and dependent on its value a different case structure is executed. This way distinction can be made between a “normal” heating operation and the maintain algorithm.

b) temperature falling

LabVIEW provides comparison functions like *greater than ?*, *greater or equal to ?*, *equal to ?*, *smaller than ?*, *smaller or equal to ?*.

In this case, however, due to the discrete and straight portions of the temperature curve, discussed before, more than just the current and the preceding value have to be compared. Otherwise, the condition would change too fast to execute a timed heating loop.

Fig. 26 *: Help window for falling2.vi*

It was, therefore, decided to compare the current TVAC value with both the preceding value and the average of the last 15 TVAC values, which is read from the *array8a.vi*. This can easily be done using two simple comparisons and an arithmetic *OR ?* function.

Instead of making these single comparisons every time they were needed, which would have used a lot of space in the CASE structures, these calculations were put into two VIs called

- *falling2.vi*: TVAC < average **or** TVAC < last value
- *smaller or equal.vi :* *TVAC* ≤ *average* **or** TVAC ≤ last value

These VIs are called everytime the direction of the slope has to be determined.

4. User Interface (GUI)

a. Checklist

After the control program had been successfully tested, some "cosmetic operations" were made on the graphical user interface. It is of great importance that the operator checks on a list of items everytime the TVAC and the program are operated. It was, therefore, decided to have a sequence of warning messages and check point items appear on the screen every time the control program is started.

The control program was placed into a large SEQUENCE structure, into the last of 5 frames. Thus the program cannot be started before the previous 4 frames containing

I. Display of dialogs

II. Input of the filename

III. Thermocouple identifiers

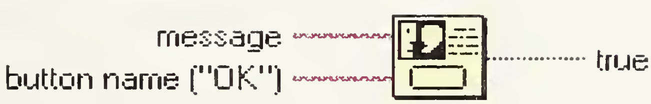
IV. Setpoint determination

V. Control program

have been executed.

I. Warning messages

A series of 4 dialog boxes is displayed, which the operator has to terminate by clicking an OK button. The function used for this task is called *one button dialog.vi.*



One Button Dialog

Displays a dialog box containing a message and a single button. Default button name is OK. output is TRUE when button is clicked on.

Fig.27 : *Help window for one button dialog*

The displayed messages make the operator check on the following:

a) Is the TVAC in an operable state ?

b) Are the six relays in their open position ?

c) Are all switches on the front panel in the *OFF* position ?

d) Is the bypass switch in the *ON* position ?

II. File to write

A dialog box appears in which the operator can toggle through the directories and determine a file to write the data to. If an existing file is chosen, the new data is appended to the file. If a new name is input, a new file is created.

The file name and path are forwarded to the control program by means of so-called "sequence locals", which can be compared to local variables.

III. Thermocouple identifiers

The names for the thermocouples are presented in an array of strings which is presented on the screen. Using a function *Two button dialog.VI* the operator is asked if he wants to keep these default names or if he wants to change them.

If he decides to change the names, the VI *changearray.vi* is executed. The string array with the default names appears and the user can modify the inputs. When the change is done, pressing the *OKAY* button closes arraychange.vi and returns to the control program.

IV. Setpoint

When the fourth sequence frame is active, the VI *setpoint.vi* is executed on the screen, saying "**Please insert the first setpoint !**". The operator is prompted to input the datapoint into a numeric control window.

Since the TVAC may not exceed the temperature range between -60 and +160 degrees, only setpoints in that range are accepted. The VI remains active until a proper setpoint is determined. The setpoint is forwarded to the last frame and the control program is started. The control algorithm immediately starts with the temperature control function that is needed to reach the setpoint temperature.

b. The Diagram

Appendix E contains the diagram of *control loop7.vi*, showing the large *SEQUENCE* structure. To simplify the structure of the control loop portion located in frame 5, two subVIs were created and substituted for the complex *CASE* scenarios of that frame:

a) **Heating.vi**

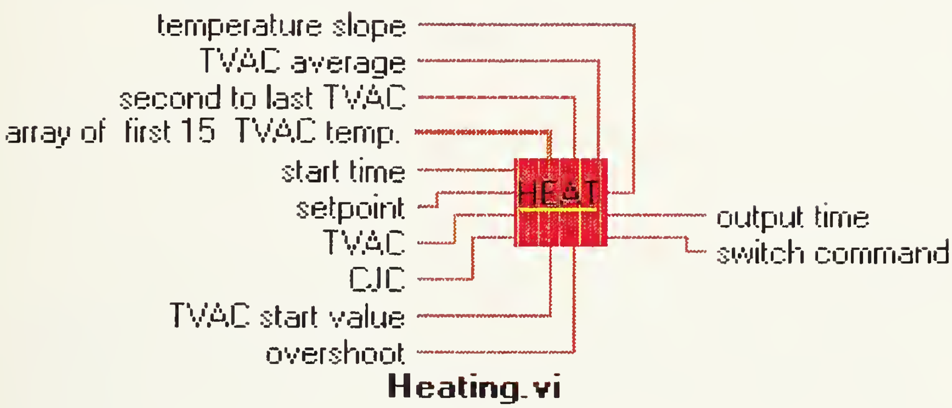


***Fig. 28** : Help window for Heating.vi*

This VI contains all the CASE structures that are needed to perform the heating and maintaining tasks. As shown in Fig. 28 above, *heating.vi* uses an amount of 10 parameters which have influence on the TVAC temperature for evaluation. According to the inputs, a heat algorithm is executed.

The VI consists of a sequence of CASE structures which work similar to the maintain loop discussed in Chapter V.B.3.c. The diagram of this VI is contained in Appendix G. Refer to the block diagram in Chapter V.B for further information on the heat algorithms.

b) **Cooling vi**

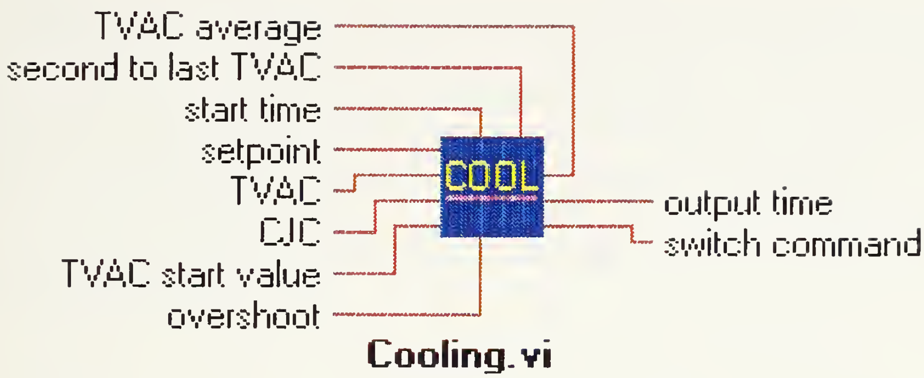


***Fig. 29** : Help window for Cooling.vi*

Cooling.vi serves as a substitute for the *CASE* structures determining the appropriate cooling mode. It contains a sequence of *CASE* structures, in which the 8 input parameters are evaluated. The necessary cooling function is determined according to the task which is to be performed and as a function of the current temperature level. *SUBZERO COOLING* is only performed below -40 °C in order to avoid putting stress onto the cooling system.

The diagram of cooling.vi is shown in Appendix H. For further information on the algorithm of this VI refer to the block diagram in Chapter 5.C.

Appendix E contains the VI diagram for the control loop program. If any further explanation of the diagram is needed refer to the LabVIEW documentation.

D. BLOCK DIAGRAMS

To show the structure of the control program, this chapter contains the block diagrams of the program and subVIs, *Cooling.vi* and *Heating.vi*. The ovals designate the algorithms that are executed. Although these actions contain the names "WAIT' and "EXECUTE MAINTAIN LOOP" several times, the algorithms are actually different and depend on the temperature level, the control mode and task to be performed.

The maintain loops are similar to the one explained in Chapter V.B.3.c, pages 58ff. For a descripton of each maintain algorithm, the reader should refer to the VI diagrams displayed in Appendices E, F, G.

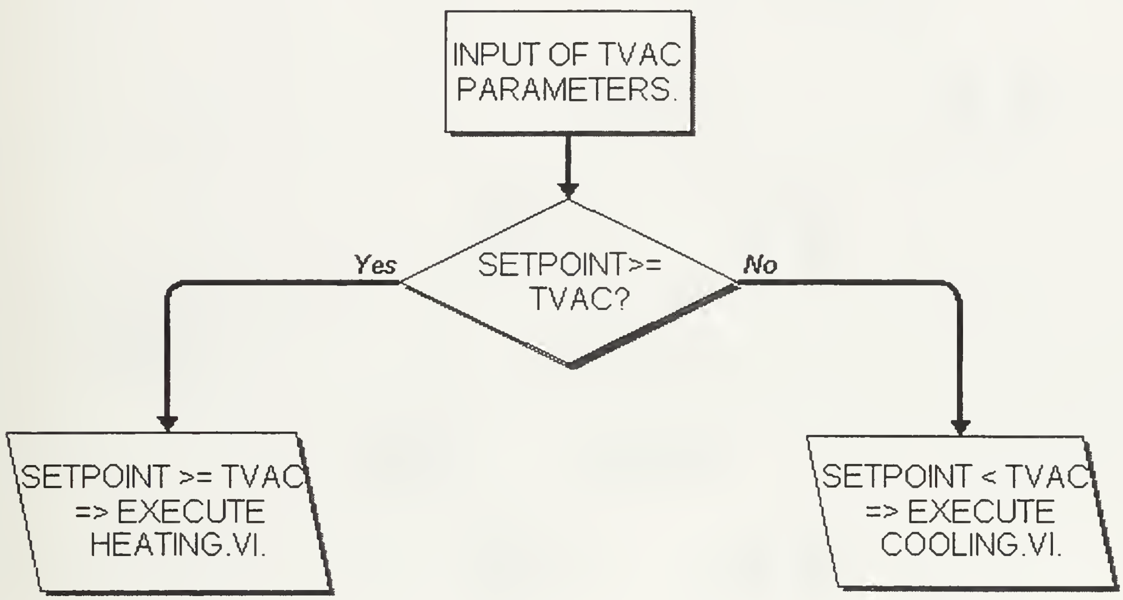


***Fig. 30** : Decision between HEATING.VIi and COOLING.VI*

The block diagrams of the subVIs *Cooling.vi* and *Heating.vi* are displayed on the following two pages. In comparing the block diagrams with the VI diagram in the Appendices, it might be helpful to consider the TRUE CASE structures as correlating to the YES case in the block diagram. Likewise, the FALSE CASE structures in the wiring diagram can be taken as the no case in the block diagram.

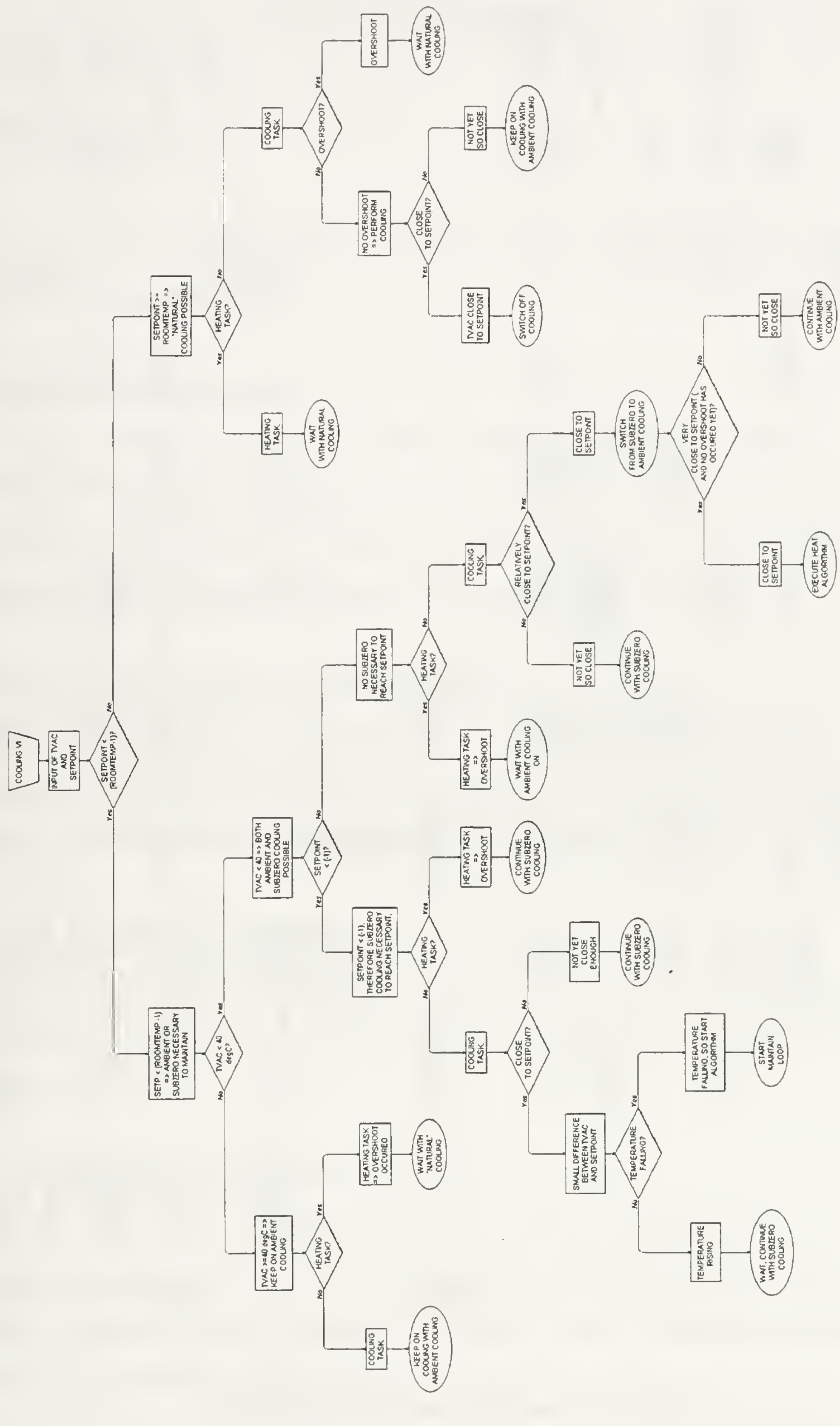


***Fig. 31**: Block Diagram for COOLING.VI*

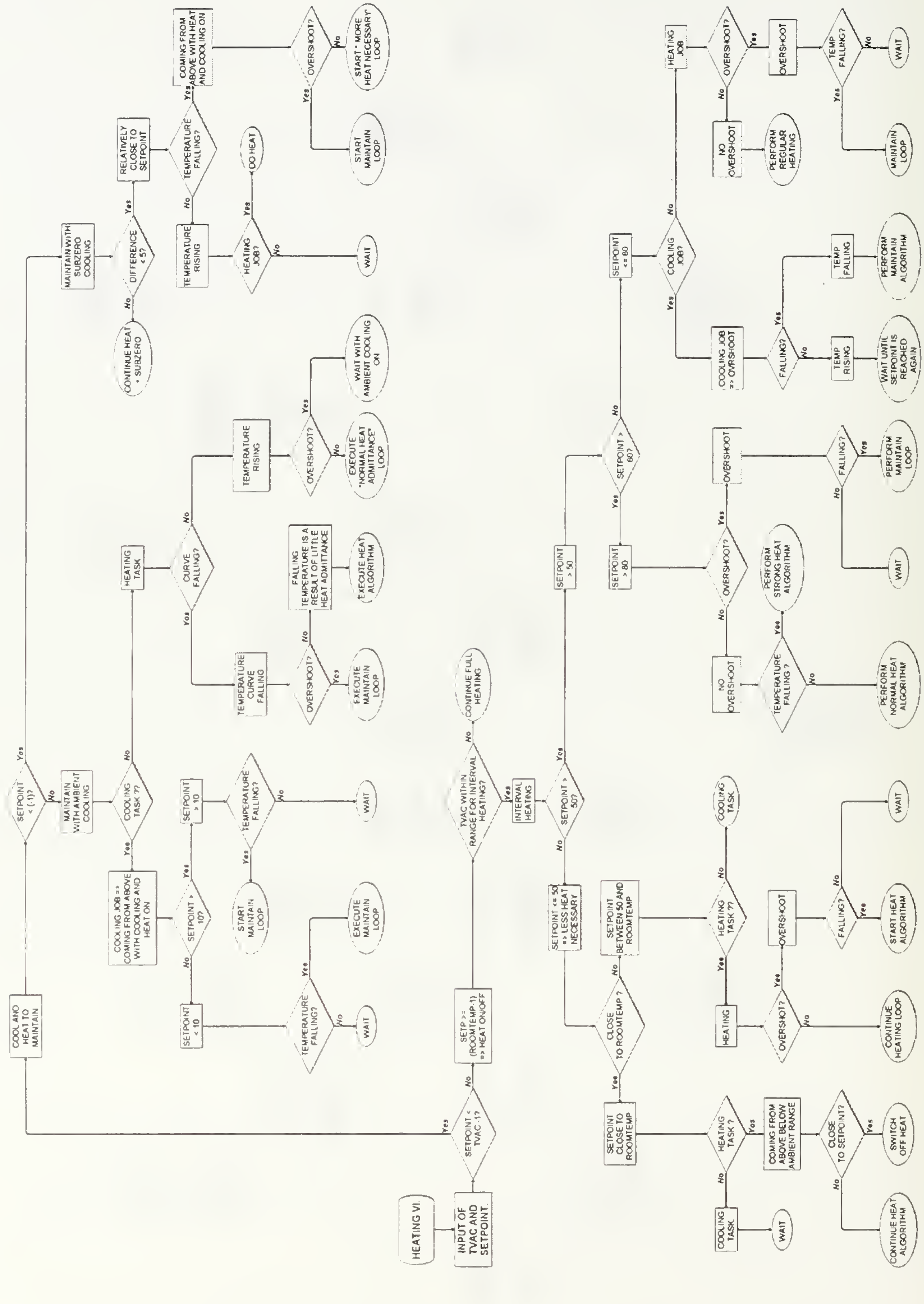


Fig. 32 : Block Diagram for HEATING.VI

VI . TVAC TESTS

A. BASICS

TVAC tests were the most time consuming portion of this project, because tests had to be performed four times :

1. At the beginning of the project :
   The TVAC characteristics had to be studied, under both vacuum and ambient pressure. Also, the way the original temperature controller, *TempTenn*, used the controls to get to a setpoint and maintain the temperature was learned. This series of tests was done by manually clicking switches on the TVAC front panel.

2. After the computer control had been established :
   Having the computer as the means of control, another series of tests had to be conducted. This time, the values for climb rates and slopes had to be taken and they had to be compared to the values obtained in the first series of tests.

3. Testing the control loop program :
   With the computer being in control, the algorithms used in the control program had to be tested and verified. Again, the values had to be compared to those of the previous tests.

4. Tests after "normal" test specifications :
   After the program had succeeded in reaching setpoints and maintaining the temperature there, a test meeting requirements for vacuum tests was performed. This time a temperature probe was put into the chamber to test, if a temperature change rate of 1°C per minute could be met.

B. RESULTS

The most important results throughout the series of more than 130 tests performed under vacuum are stated in the tables below.

1. HEATING :

Climb rates :

Maximum climb rate with full HEAT on :	≈ 11.3 °C per minute
Average climb rate with full HEAT on :	≈ 9.4 °C per minute
Climb rate with 2 minutes HEAT on, 1 minute off:	≈ 6.2 °C per minute
Climb rate with 1 minute HEAT on, 1 minute off:	≈ 4.8 °C per minute
Climb rate with 1 minute HEAT on, 2 minutes off:	≈ 1.8 °C per minute
climb rate with ambient cooling on :	≈ 3.0 °C per minute
climb rate with sub-zero on :	≈ 1.9 °C per minute
natural climb rate from sub-zero temperatures:	≈ 0.3 °C per minute
Average climb rate for heating between TVAC extreme temperatures (-78 -> +182)	≈ 5.5 °C per minute

Overshoot :

Longest time for overshoot after full heating	4 minutes 12 seconds
Shortest time for overshoot after full heating	3 minutes 28 seconds
Temperatures:	
Greatest overshoot after full heating:	12 °C
Smallest overshoot after full heating :	28 °C

The highest temperature that can be reached with the Space jr. TVAC is 182.5 °C, and only by overshooting, because an electronic cutout takes effect when the TVAC reaches a temperature higher than 175 °C.

2. COOLING

The cooling is performed with a negativ slope which increases as a function of the time and as a function of the minimum temperature. The decay seems to be exponential. Since it is difficult to determine a rate of decrease, the cooling efficiency is presented in some examples.

Natural cooldown:

Fastest decrease	- 4.50 °C per minute
Slowest decrease	- 0.08 °C per minute
Lowest temperature reached :	23 °C (room temperature)

AMBIENT COOLING:

Fastest decrease	- 3.8 °C per minute
Slowest decrease	- 0.03 °C per minute
Lowest temperature reached :	-10.12 °C

SUBZERO COOLING:

Fastest decrease	- 2.50 °C per minute
Slowest decrease	- 0.04 °C per minute
Lowest temperature reached :	-74.5 °C

Appendix B contains some Microsoft Excel charts which display some of the temperature profiles that were performed throughout the tests.

Appendix C shows selected screenshots taken from the VIs during tests.

VI. RECOMMENDATIONS AND CONCLUSION

A. RECOMMENDATIONS

For reasons dicussed in this thesis, the TVAC cannot be controlled using the SCXI system within a range of only 1 degree. The main problem is, that maintaining a temperature can only be done by admitting heat and not by also switching cooling on and off. Furthermore, heat is always admitted in the same amount, it is either on or off.

It is recommended that a more diversified means of heating be installed. If heat was admitted with variable intensity, compared to the same amount of time, and if the heat admittance was tuned with the heating job which is to be performed, control could be made much more exact and temperatures could be maintained within a smaller range.

The analysis of the parameters influencing the process is only as exact as the boundary conditions. It would be useful to perform a large series of tests, in which the heating efficiency and the cooling efficiency are determined as functions of the temperature level. Especially maintaining processes at subzero temperatures suffer in accuracy because the cooling system changes efficiency with temperature.

The control of the modified TVAC is much more exact than with the TempTenn controller that was originally installed in the TVAC. Due to the *data flow principle* of LabVIEW a single iteration of the program takes more than 1 second, which makes the temperature slope rather unpredictable. The charts in Appendix C display the way the data acquisition is performed, including all the straight parts in the temperature chart. Further studies on making the curve more continuous are recommended. Higher accuracy could be achieved if a **real** adjustment of heat admittance to the slope could be accomplished.

Although the reading of the chamber pressure was one of the goals of this thesis project, it could not be added to the data acquisition because some parts necessary to connect the Granvillle-Phillips pressure gauge controller to the SCXI system were not

available. The connection has to be made later. Reading the pressure could bring an additional improvement to the temperature control algorithm because the TVAC temperature is a function of the chamber pressure.

B. CONCLUSION

The computer instrumentation decribed in this thesis meets all the requirements that were set up before the project was started, except for the pressure reading. A very exact means to control the Tenney Space jr. is provided, although it is dependent on a large number of continuously changing parameters.

The design changes which were performed on the TVAC housing extend the operational profile by providing connectors to actually operate components that are to be tested throughout the tests. Even the additional amount of heat, resulting from energy production in the subsystem, can be taken into account.

The fact that the flow control of liquid nitrogen and starting the recirculator still have to be done manually keep the system from being fully computer controlled, but this seems acceptable to avoid catastrophic failures.

LIST OF REFERENCES

NATIONAL INSTRUMENTS, *LabVIEW V3.1.1 Documentation*, September 1994

Contents :

I. LabVIEW Function Reference Manual

II. LabVIEW for Windows User Manual

III. LabVIEW for Windows Tutorial

IV. LabVIEW Base Analysis VI Reference Manual

V. LabVIEW Data Acquisition VI Reference Manual for Windows

VI. LabVIEW Instrument I/O Reference Manual for Windows

VII. LabVIEW Master Index

NATIONAL INSTRUMENTS, *SCXI Documentation*, May 1994

Contents :

I. Getting Started with SCXI

II. AT-MIO E Series User Manual

III. SCXI-1100 User Manual

IV. SCXI-1120 User Manual

V. SCXI-1161 User Manual

VI. SCXI-1000/1000DC/1001 User Manual

SPACE SYSTEMS ACADEMIC GROUP, *Solar Panel Thermal-Vacuum Test Manual*, NAVAL POSTGRADUATE SCHOOL, 1993

SPACE SYSTEMS ACADEMIC GROUP, *PANSAT Functional Requirements Document*, NAVAL POSTGRADUATE SCHOOL, 1993

V.L.Pisacane and R.C.Moore, *Fundamentals of Space Systems,* OXFORD UNIVERSITY PRESS 1994

NATIONAL INSTRUMENTS, *Measuring Temperaure with Thermocouples*, Application Note 043, June 1993

VARIAN, *Instruction Manuals*, VARIAN, January 1991

Contents :

I. Turbo V-200 Controller

II. Turbo V-200 Pumps

TENNEY, Instruction Manual for the Space Jr. Thermal Vacuum Space Simulator

Appendix A

Determination of the appropriate cooling method

Cooling :

TVAC temperature	**Natural cooling**	**AMBIENT COOLING**	**SUBZERO COOLING**
TVAC > 40°C	possible	possible and faster than natural cooling	not recommended because mechanical parts are put under stress
40°C > TVAC > -10°C	only possible until room temperature is reached (slow !)	possible only above -10 °C, and very slow below 0 °C	possible and relatively fast, therefore it is switched from SUBZERO to AMBIENT if TVAC is close to setpoint
TVAC < -10°C	not possible	not possible	possible until minimum temperature of -75,8 °C is reached

Maintaining :

TVAC	**HEAT**	**AMBIENT COOLING**	**SUBZERO COOLING**
TVAC > Ambient temperature	**ON**	*OFF*	*OFF*
Ambient > TVAC > -1 °C	**ON**	**ON**	*OFF*
TVAC < -1 °C	**ON**	*OFF*	**ON**

Appendix B

Importing Datafiles into Microsoft Excel

1. Open Microsoft Excel 5.0

   (Icon can be found in the Microsoft Office Group)

2. Go to *File* -> *Open*

   Choose the file you want to open in the File Dialog Window.

3. Press OK button => **Text Import Wizard** is started.

   a) In the field " Original Data Type

      Chose the file type that best describes your data "

      select *Fixed Width* .

   b) Press *NEXT* > button => *Field Width Window* appears.

      Delete the break line between the time column and the AM/PM column.

      Press *NEXT*> button .

   c) Chose columns that are not to be imported and switch their Data Format to

      " *Do not import column (Skip)* ".

      Attention : Between each thermocouple column is an unwanted column.

      Delete them before proceding. !

   d) Change the Data Format of the very first column (Date) to " *Date : MDY* ".

   e) Press " *Finish !* " button.

4. The data appears, separated into columns, in an Excel spreadsheet.

5. Click on the first data cell (A1). Go to *Insert* -> *Rows*. An empty row is inserted on top of the data cells.

6. Insert the names for the data columns. They are saved in the following order : Date, Time, TVAC, Cold Trap, SUBZERO, AMBIENT, HEAT, DOORHEAT, Relay #6, Setpoint, Cold Junction Channel, Thermocouples 1 to 18.

   Attention: Take into respect if any data column was skipped throughout the Text Import Wizard !

7. Highlight the data that is to be presented in a chart.

   Go to *Insert -> Chart ->As new sheet.*

8. Chart Wizard starts (End each of the 5 steps by pressing the "*Next*>" button.)

   Step 1 of 5 : Do not change anything

   Step 2 of 5 : Chose chart type: XY (Scatter)

   Step 3 of 5 : Chose format No.6

   Step 4 of 5 : Determine,

   - if a legend is to be displayed
   - that the data series is in columns
   - which row contains the Xdata
   - which column is used for the legend text

   Press the " *Finish !*" button.

7. The chart is displayed on the screen :

   The switch setting is represented by 5 individual lines, divided by 5 degrees each, whose values differ only by one degree. The higher value means "on", the lower means "off".

8. For further information refer to the help function (press F1) or the Microsoft Excel documentation.

Appendix C

Microsoft Excel Charts

Microsoft Excel charts :

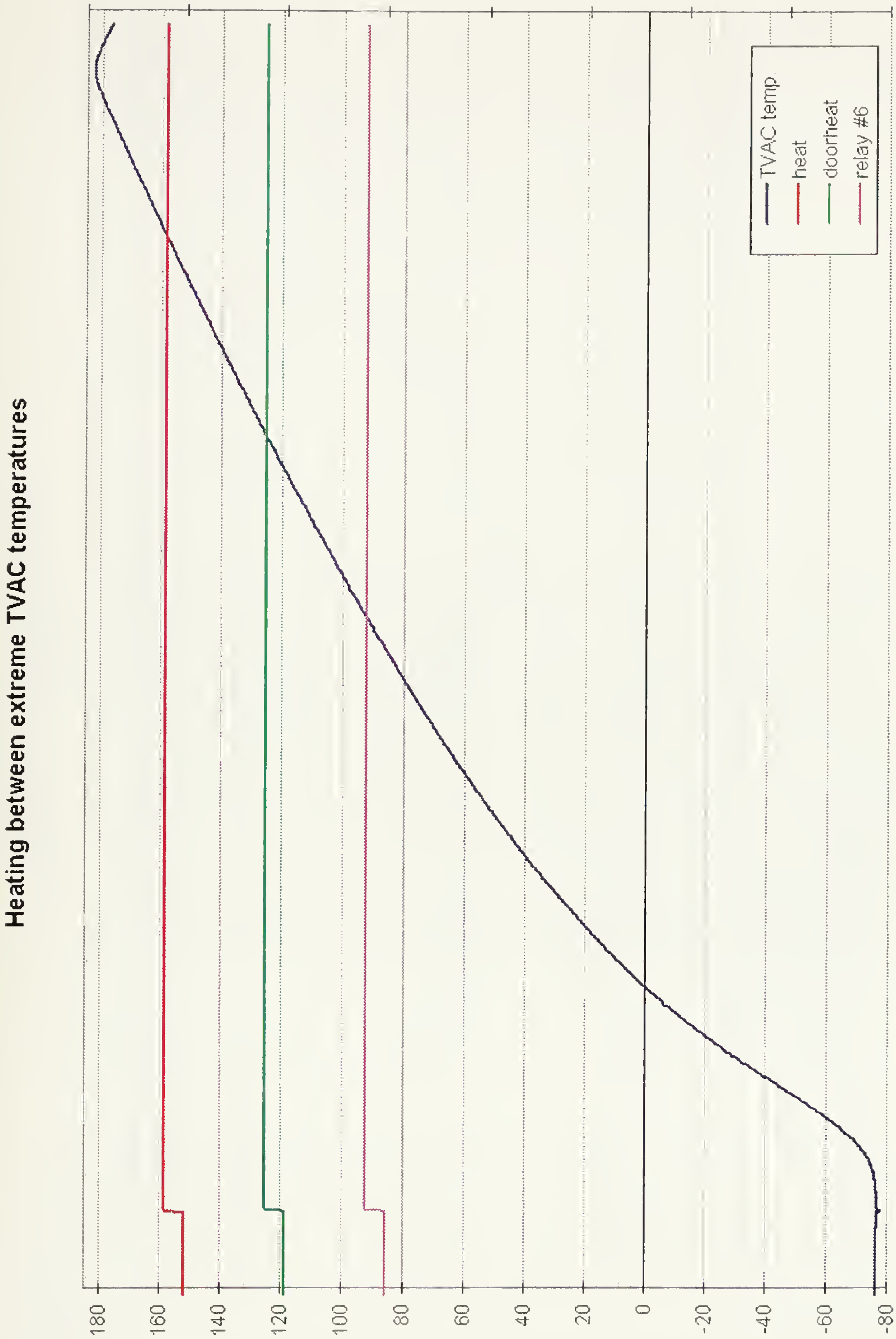


Fig. 33 : *Heating between extreme TVAC temperatures*

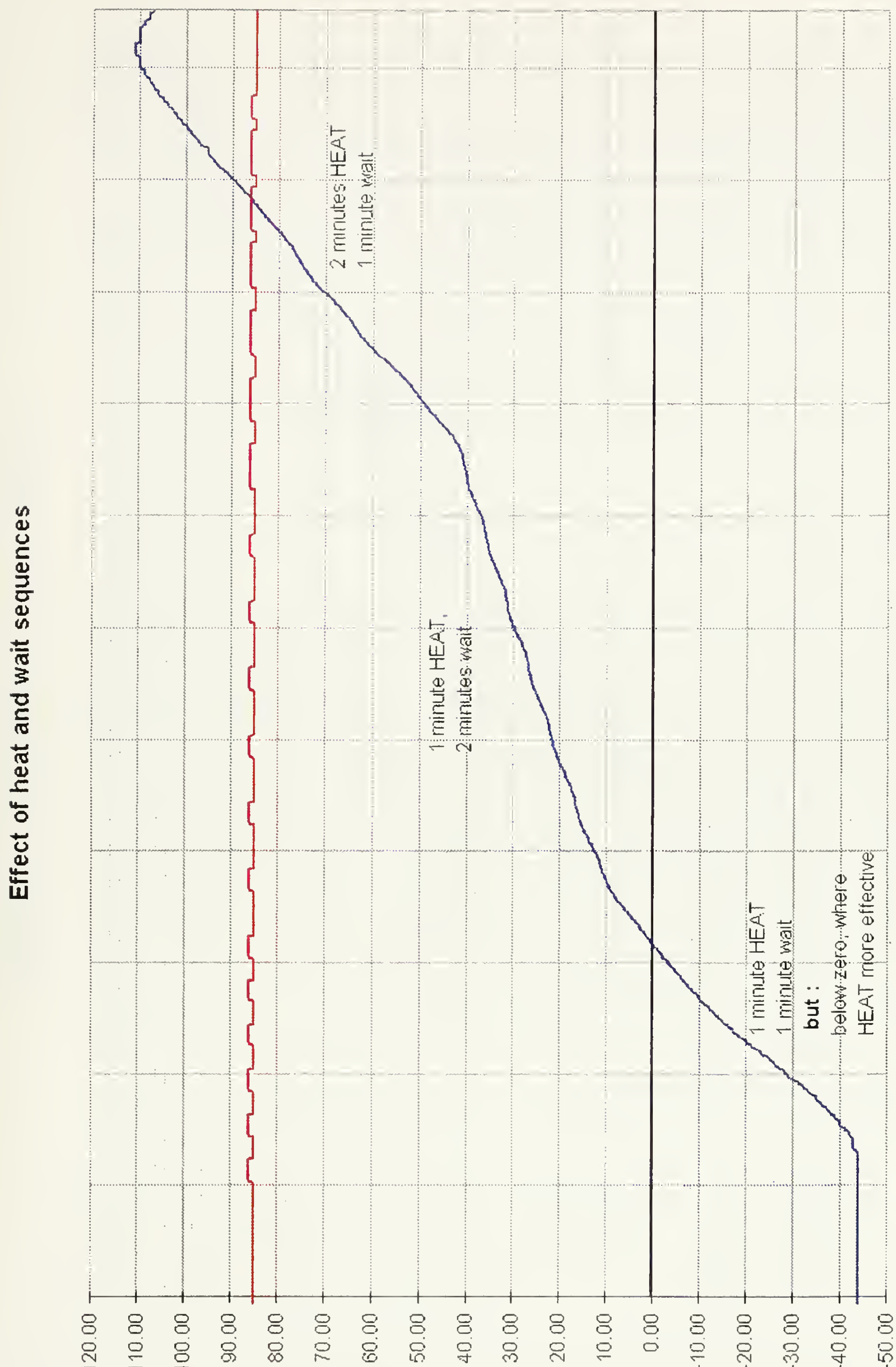


Fig. 34 : Effect of HEAT and WAIT sequences

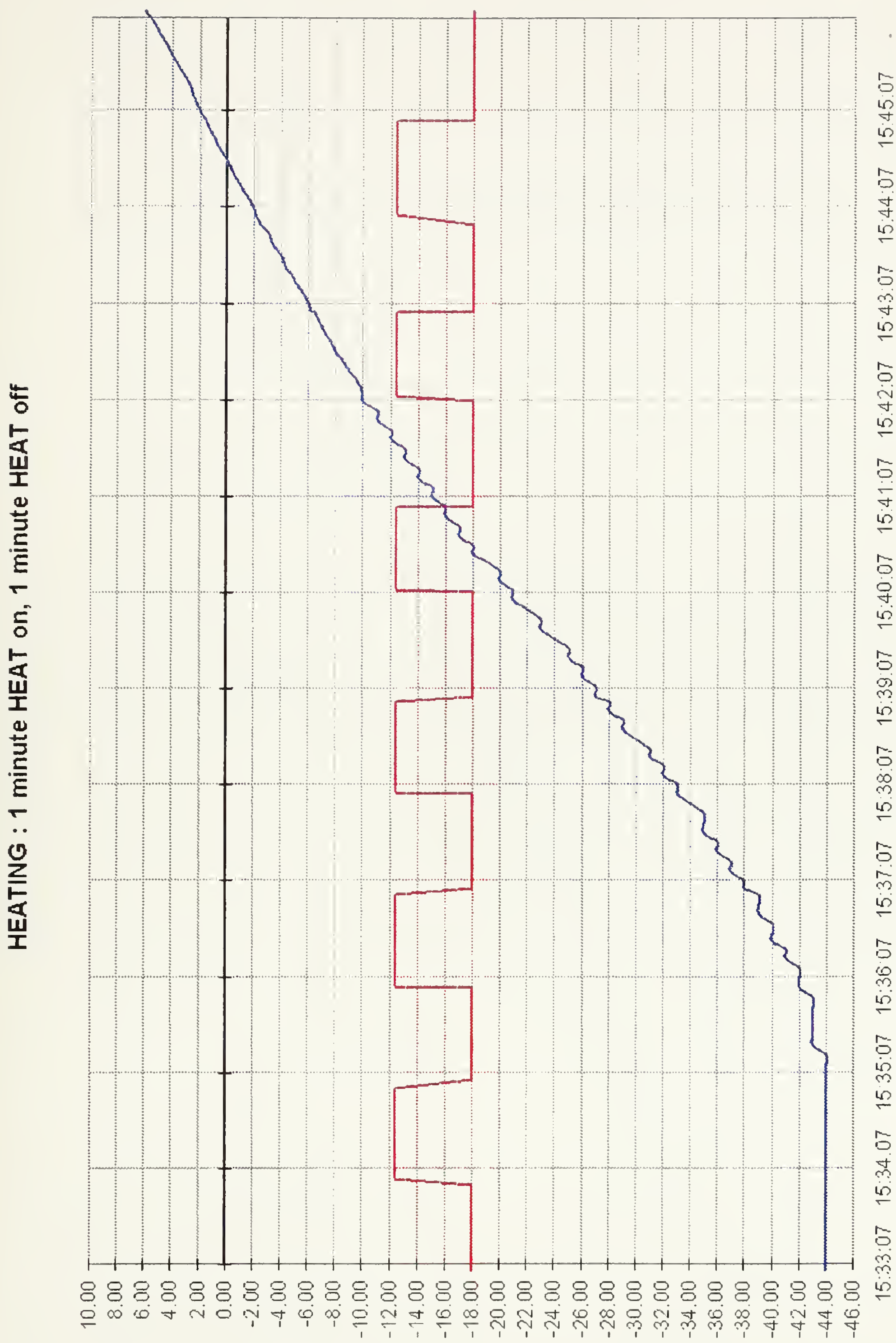


Fig. 35 : 1 minute HEAT on, 1 minute HEAT off

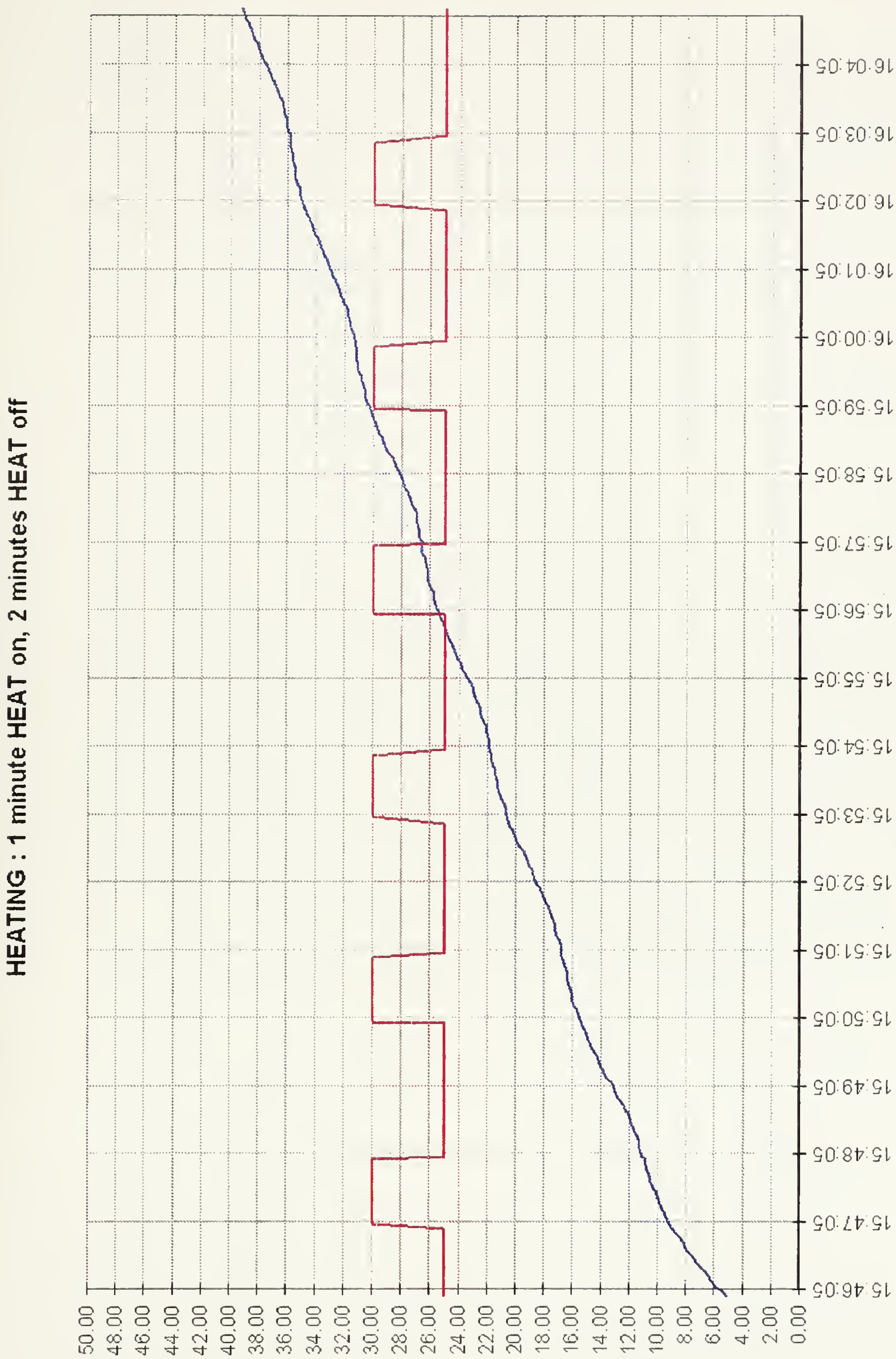


***Fig. 36** : 1 minute HEAT on, 2 minutes HEAT off*

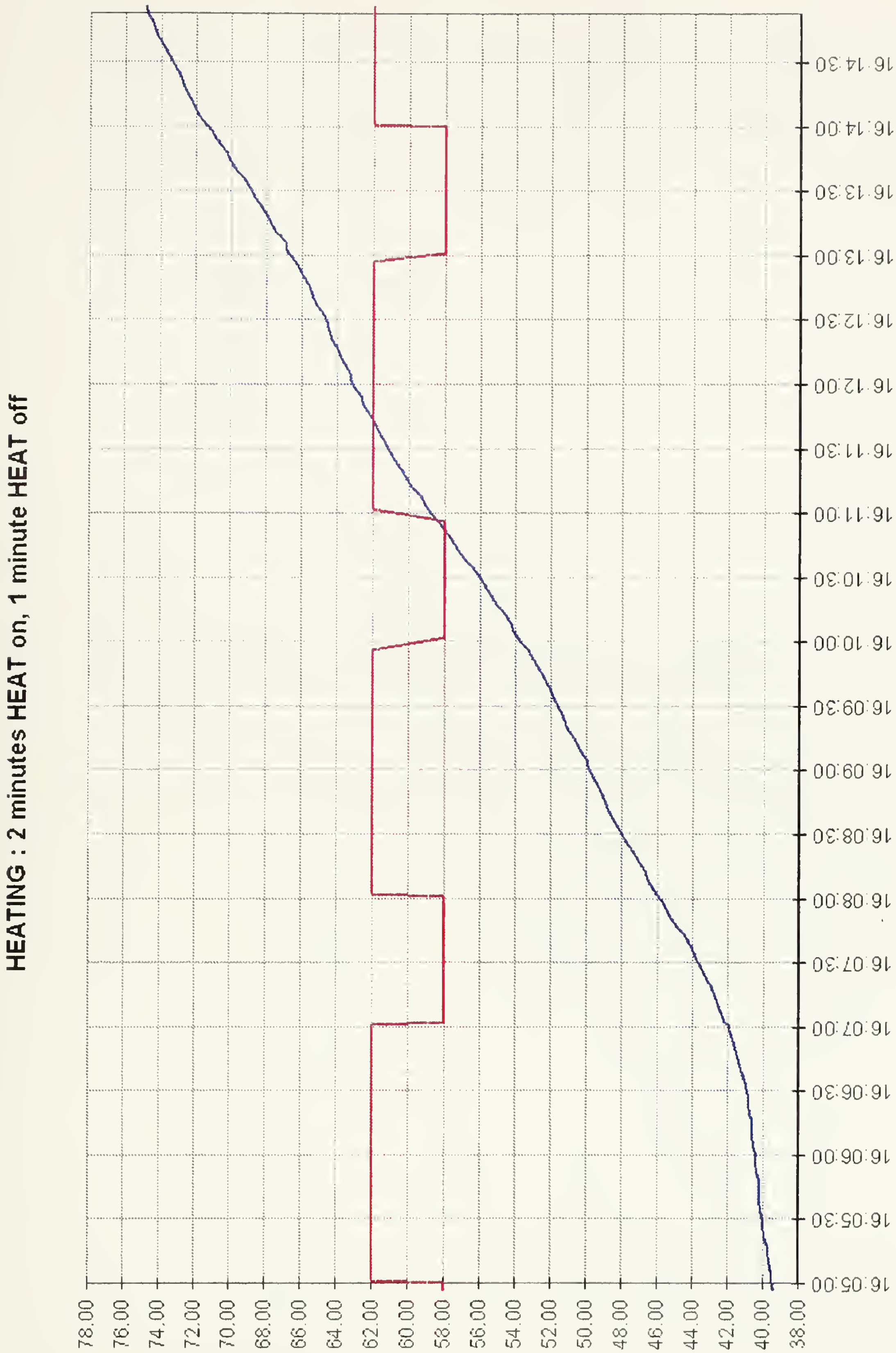


Fig. 37 : 2 minutes HEAT on, 1 minute HEAT off

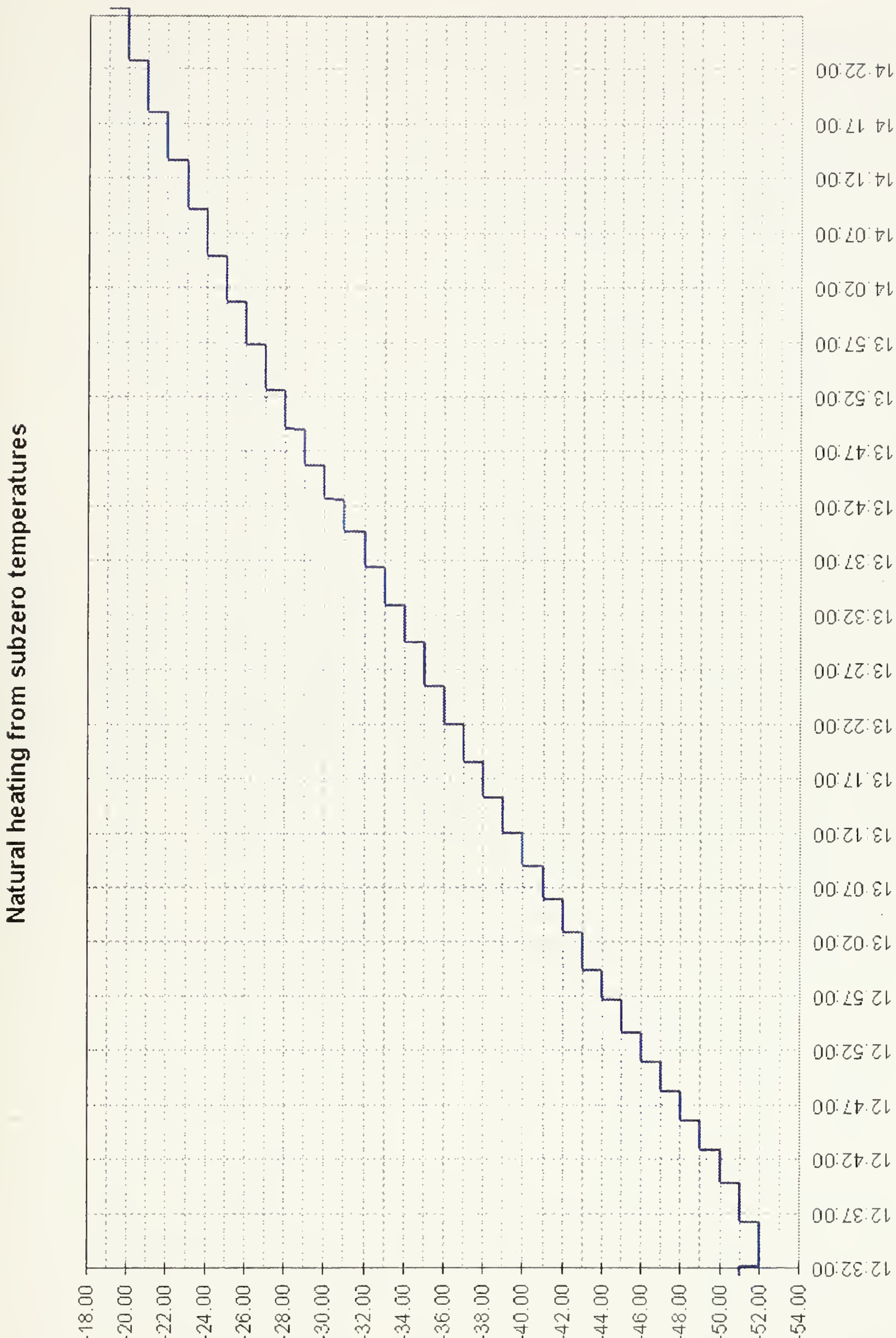


***Fig. 38** : Natural reheating from subzero temperatures*

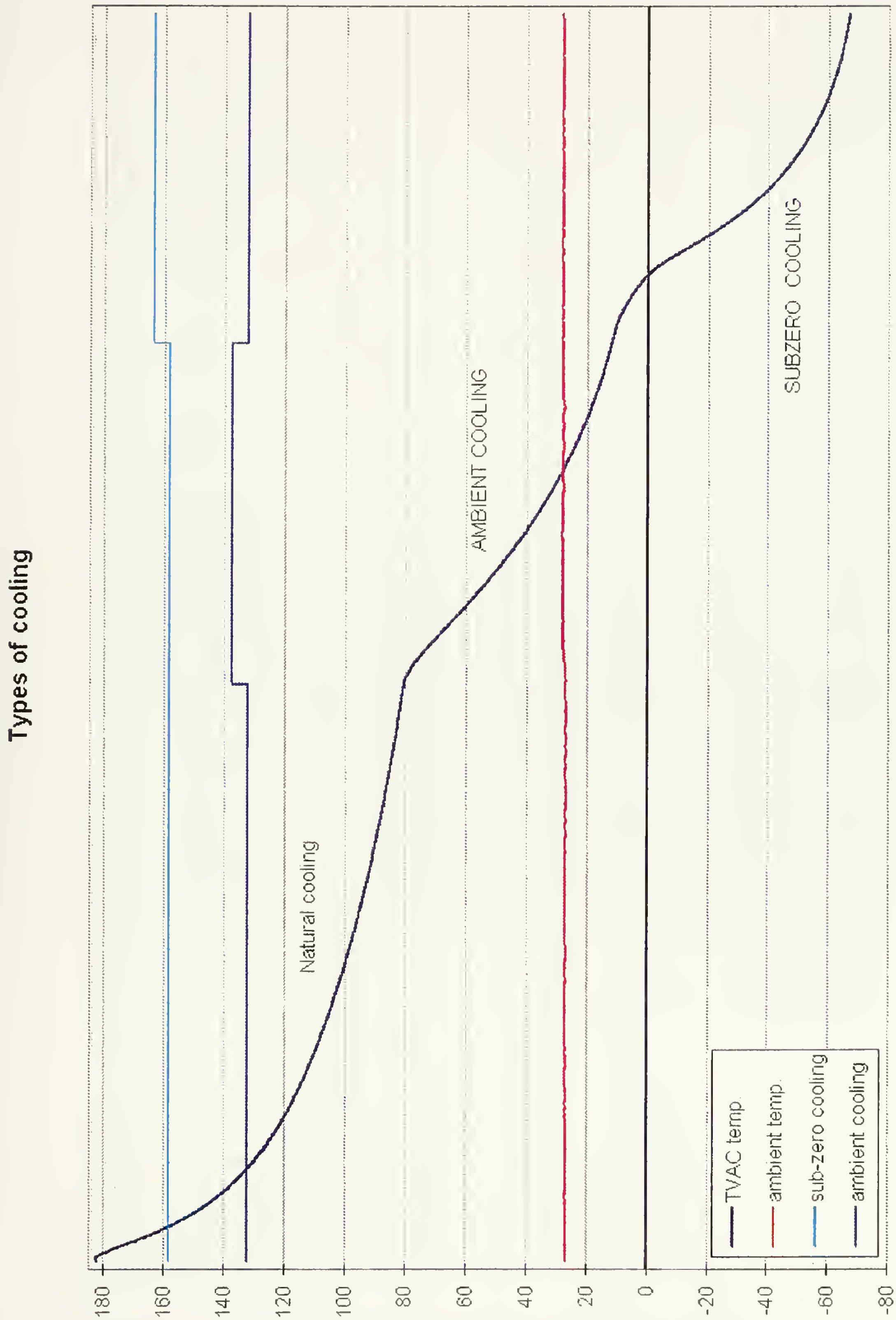


***Fig. 39** : Types of Cooling*

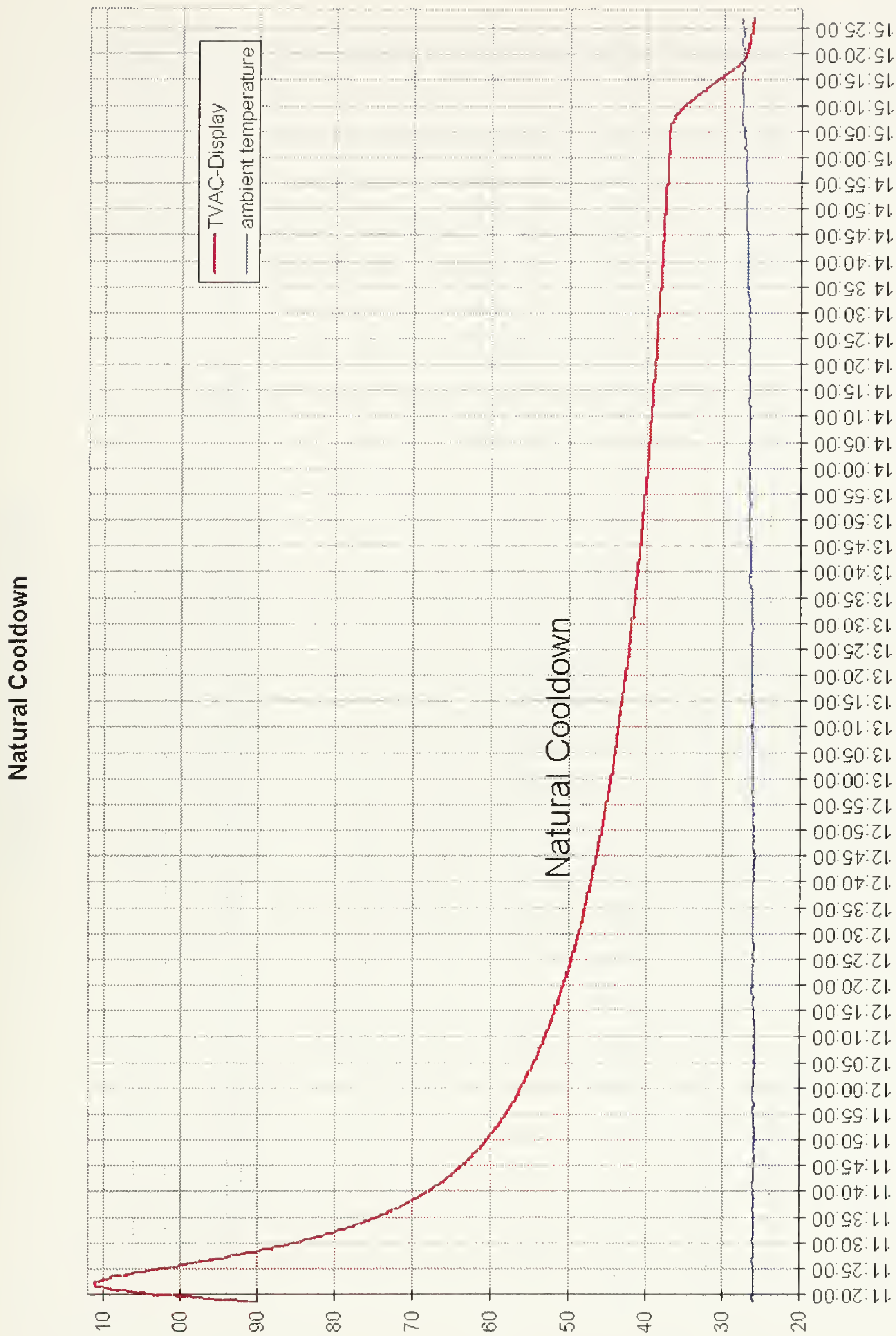


Fig. 40 : Natural Cooldown

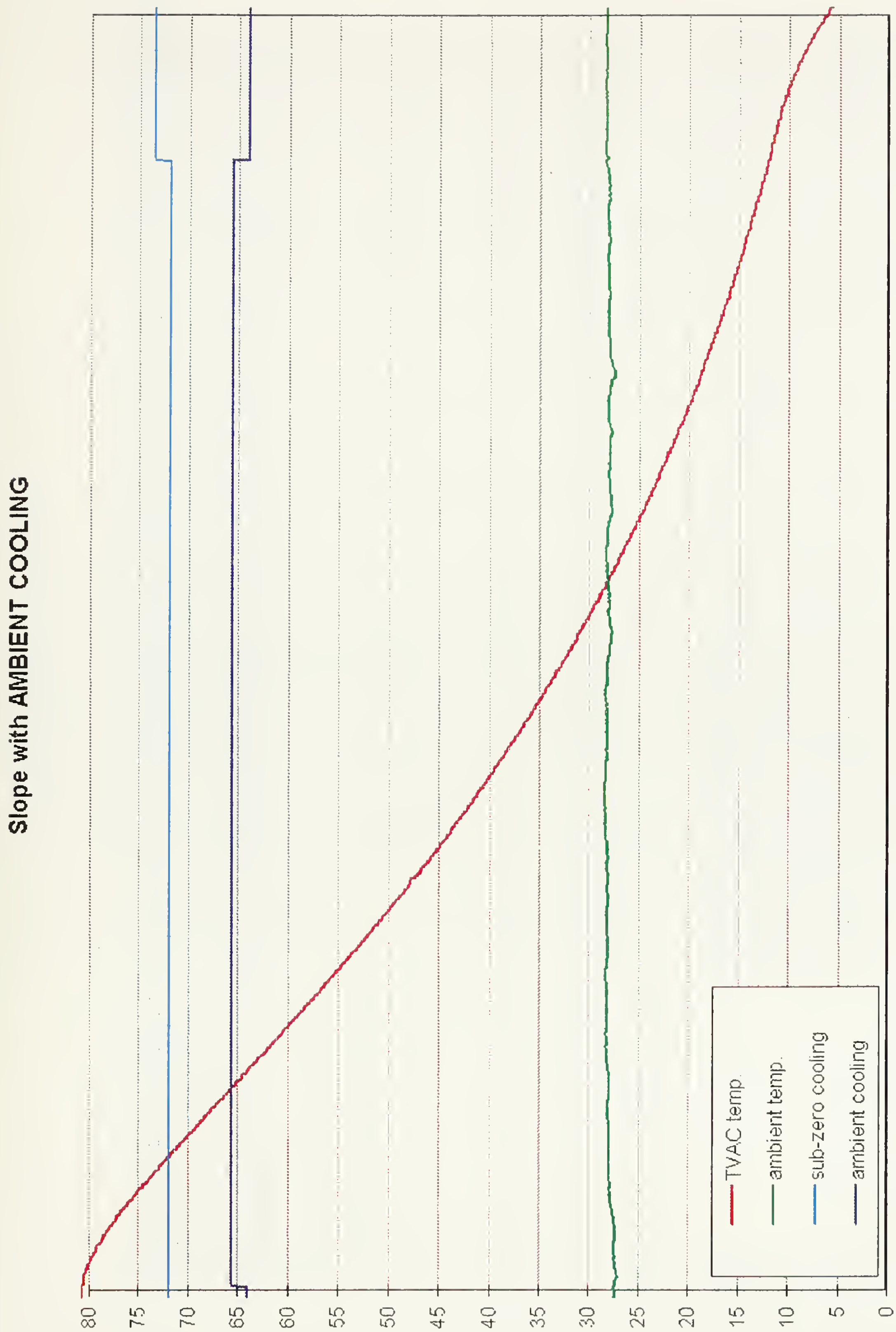


***Fig. 41** : Slope with AMBIENT COOLING*

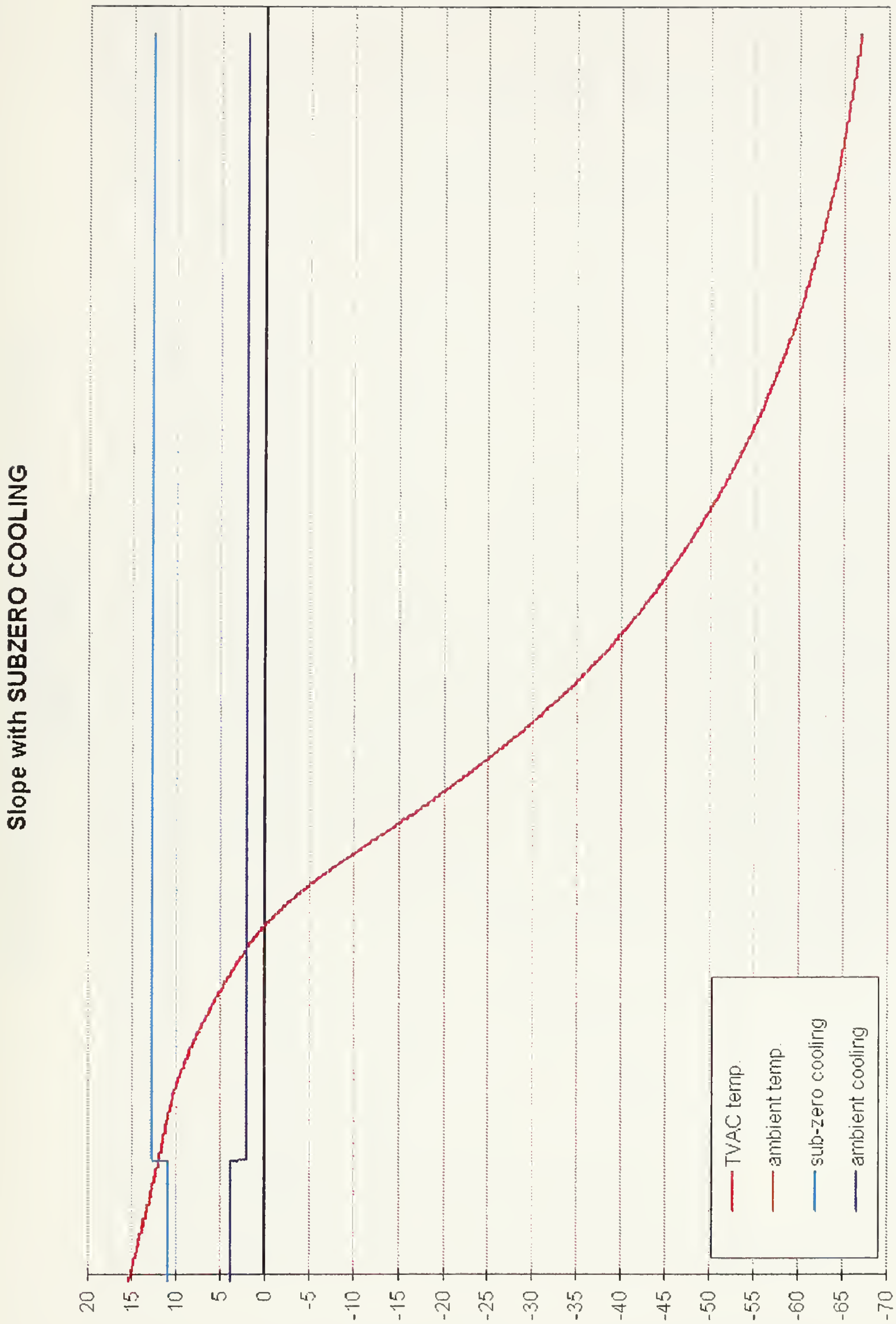


Fig. 42 : Slope with SUBZERO COOLING

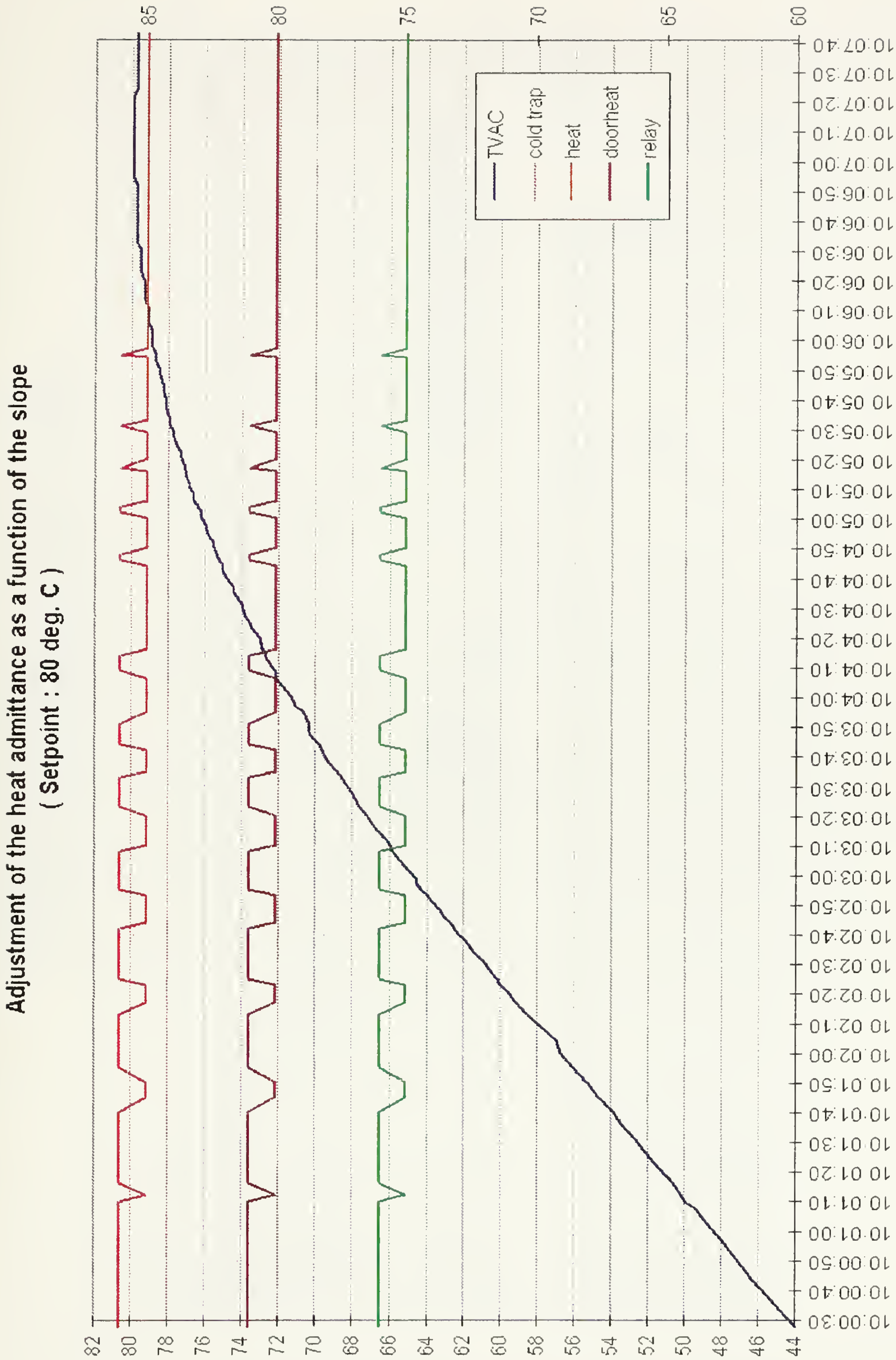


Fig. 43 : Adjustment of the heat admittance as a function of the slope

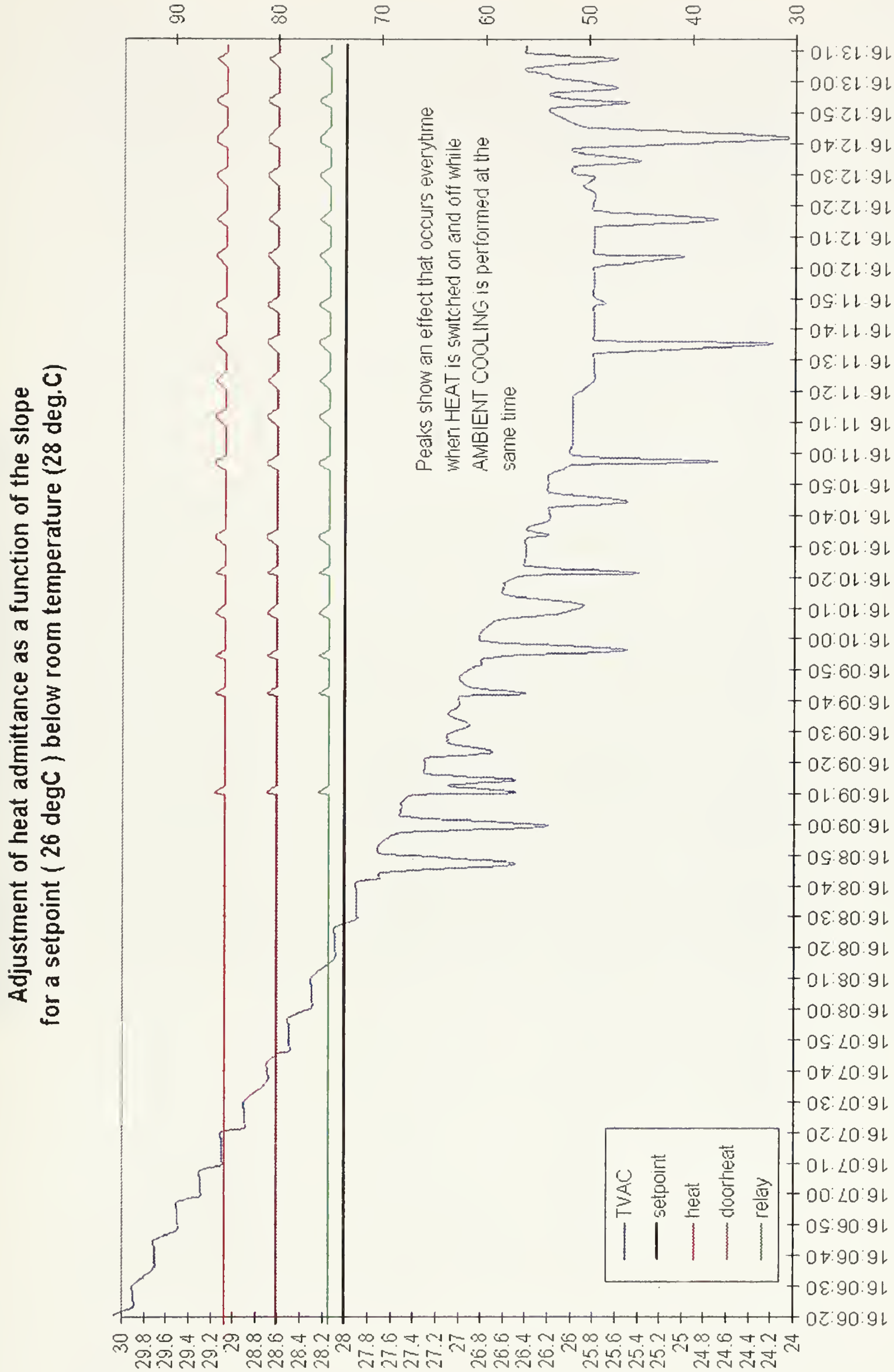


Fig. 44 : Adjustment of heat admittance as a function of the slope (with AMBIENT COOLING performed at the same time)

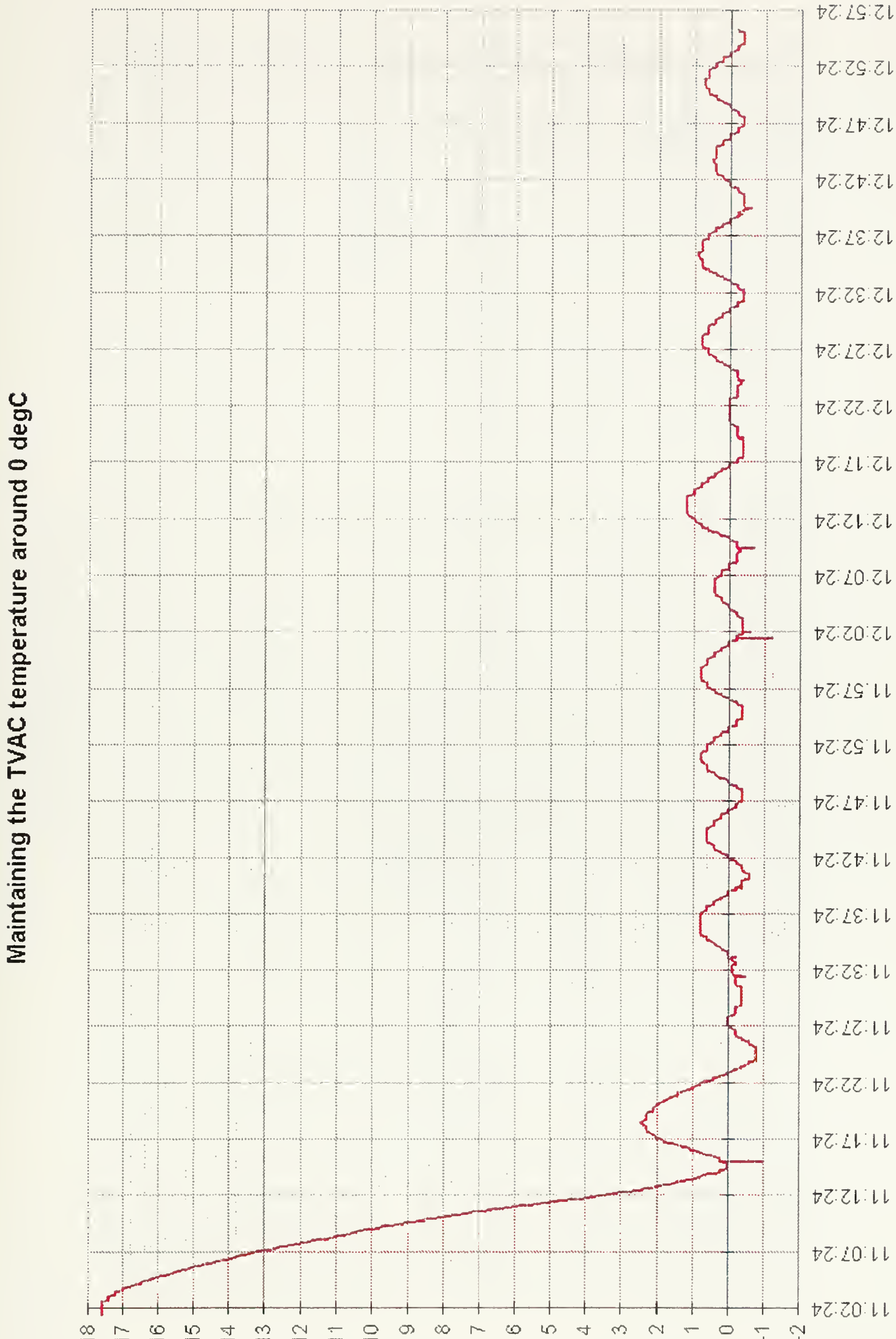


Fig. 45 : Maintain at 0°C

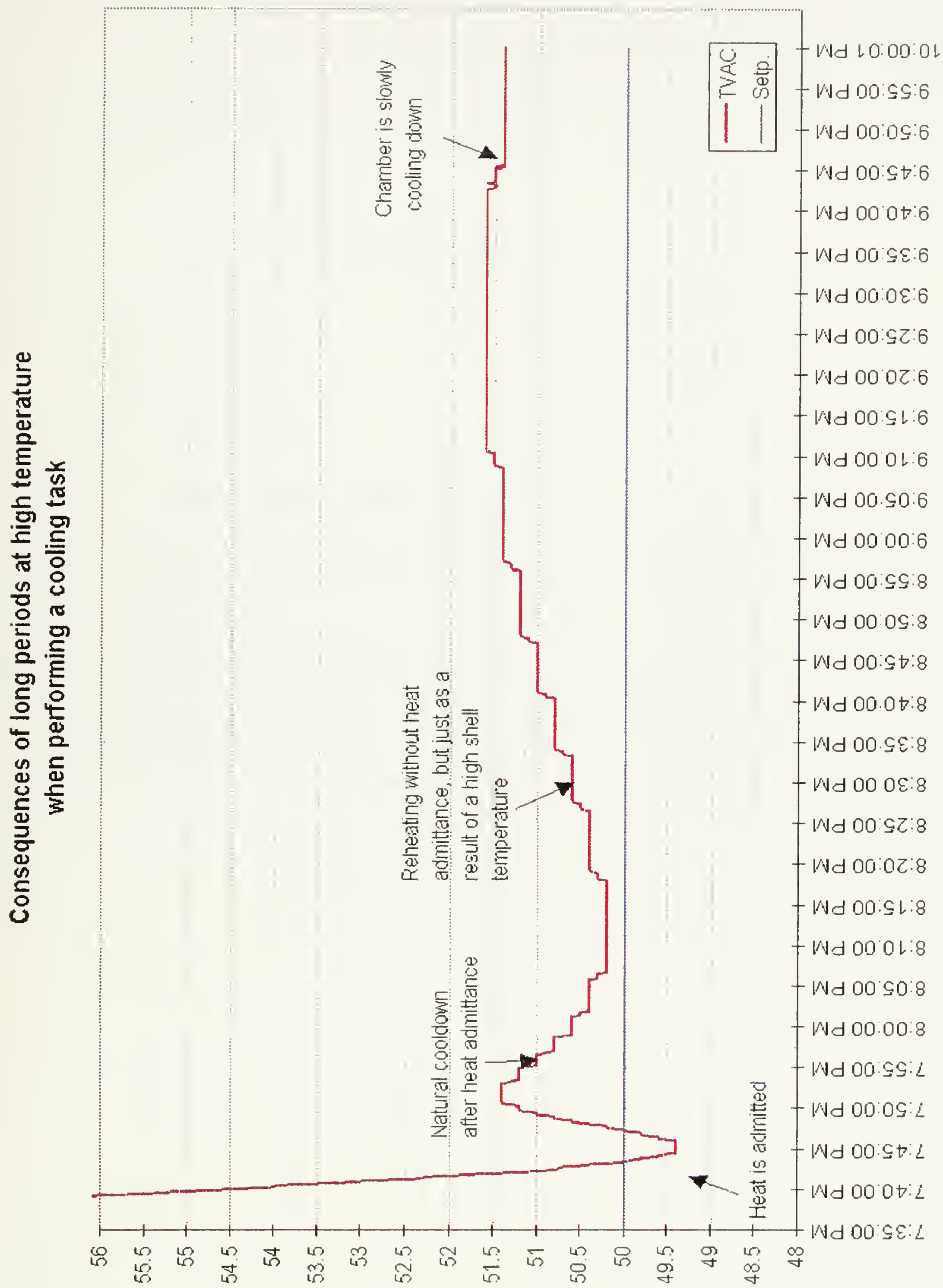


Fig. 46 : Consequences of long periods at high temperatures

APPENDIX D
SCREENSHOTS FROM THE VI

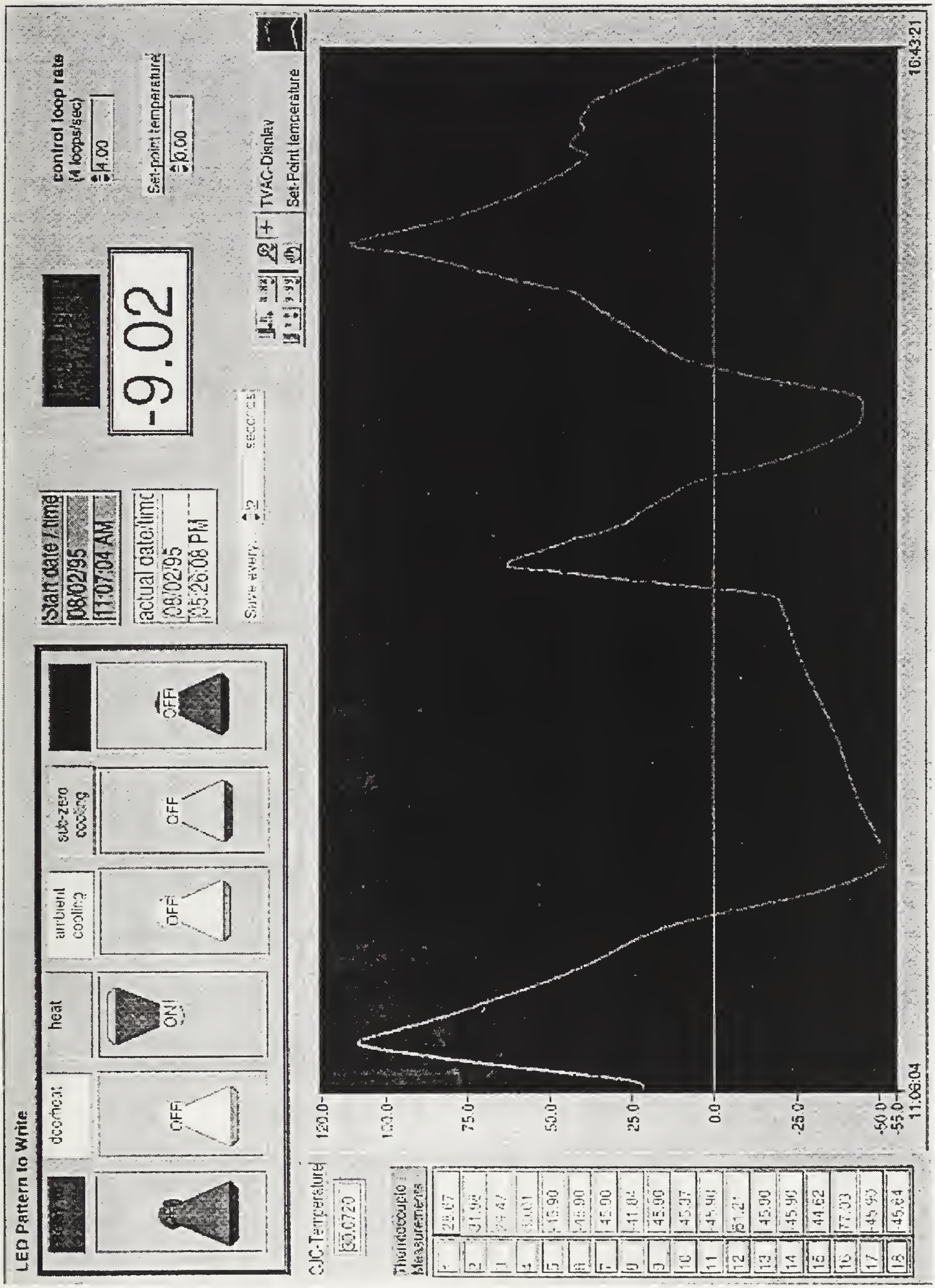


Fig. 47 : Temperature Profile

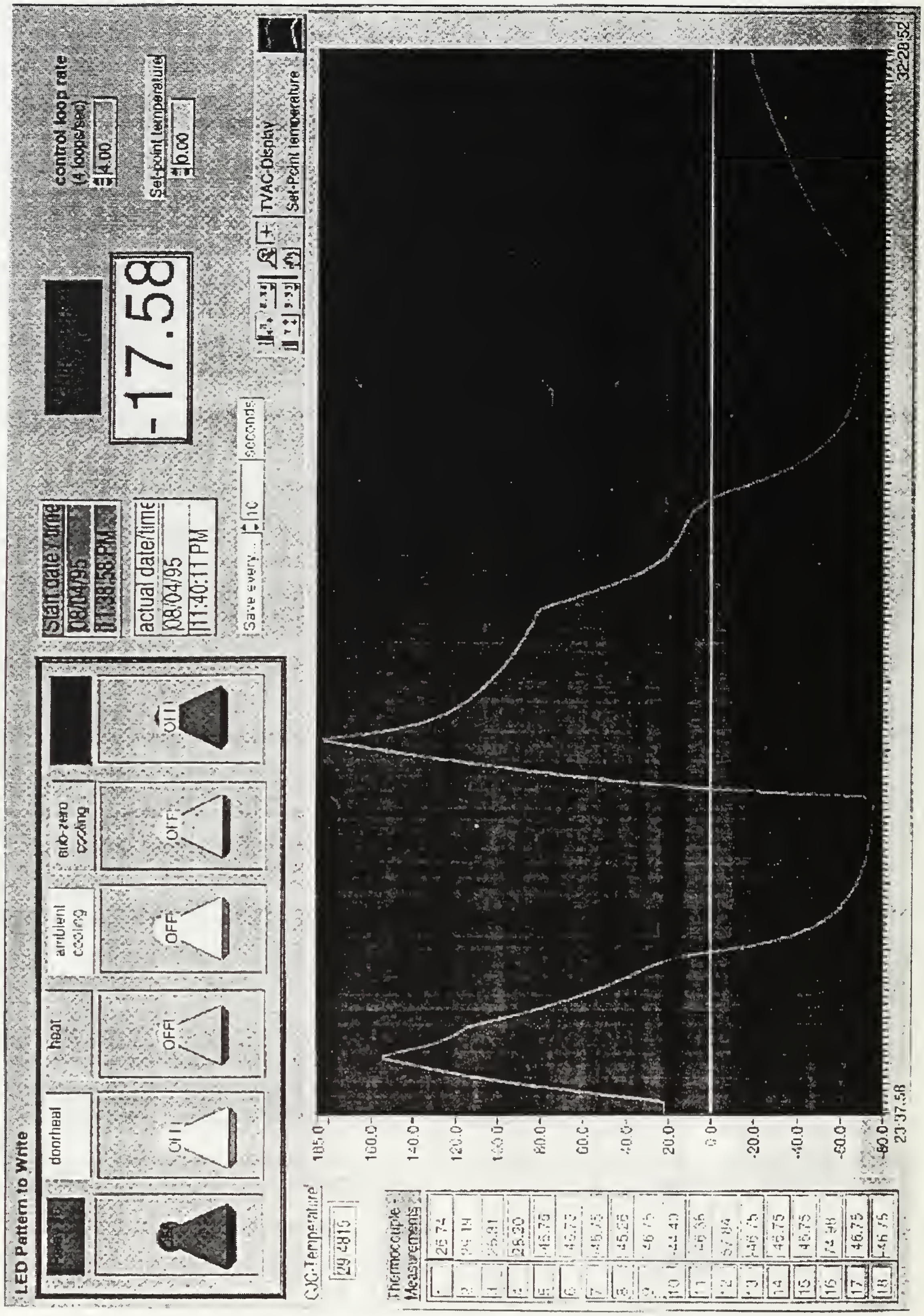


Fig. 48 : *Heating and Cooling slopes*

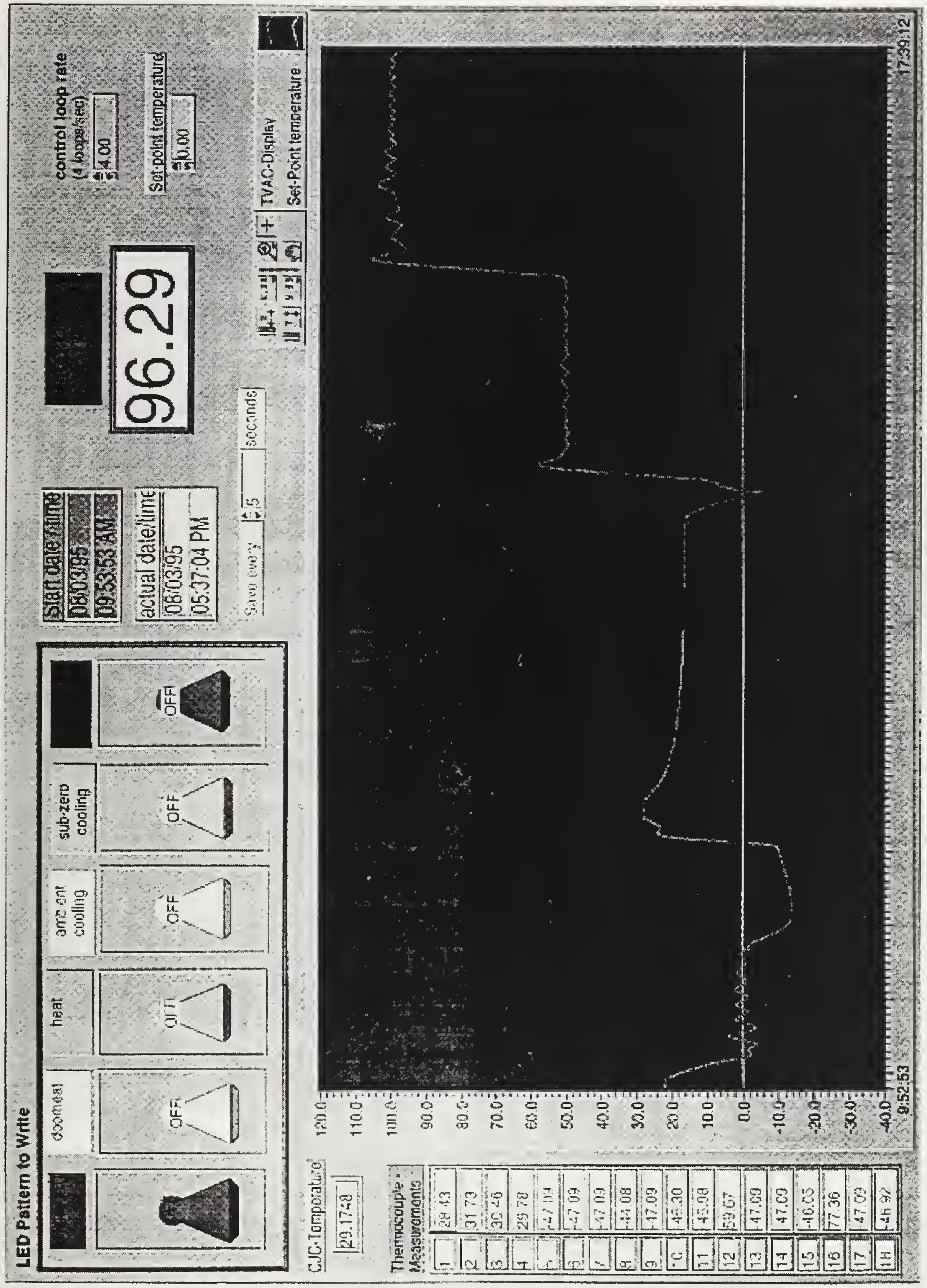


Fig. 49 : *Trials to maintain different temperature levels*

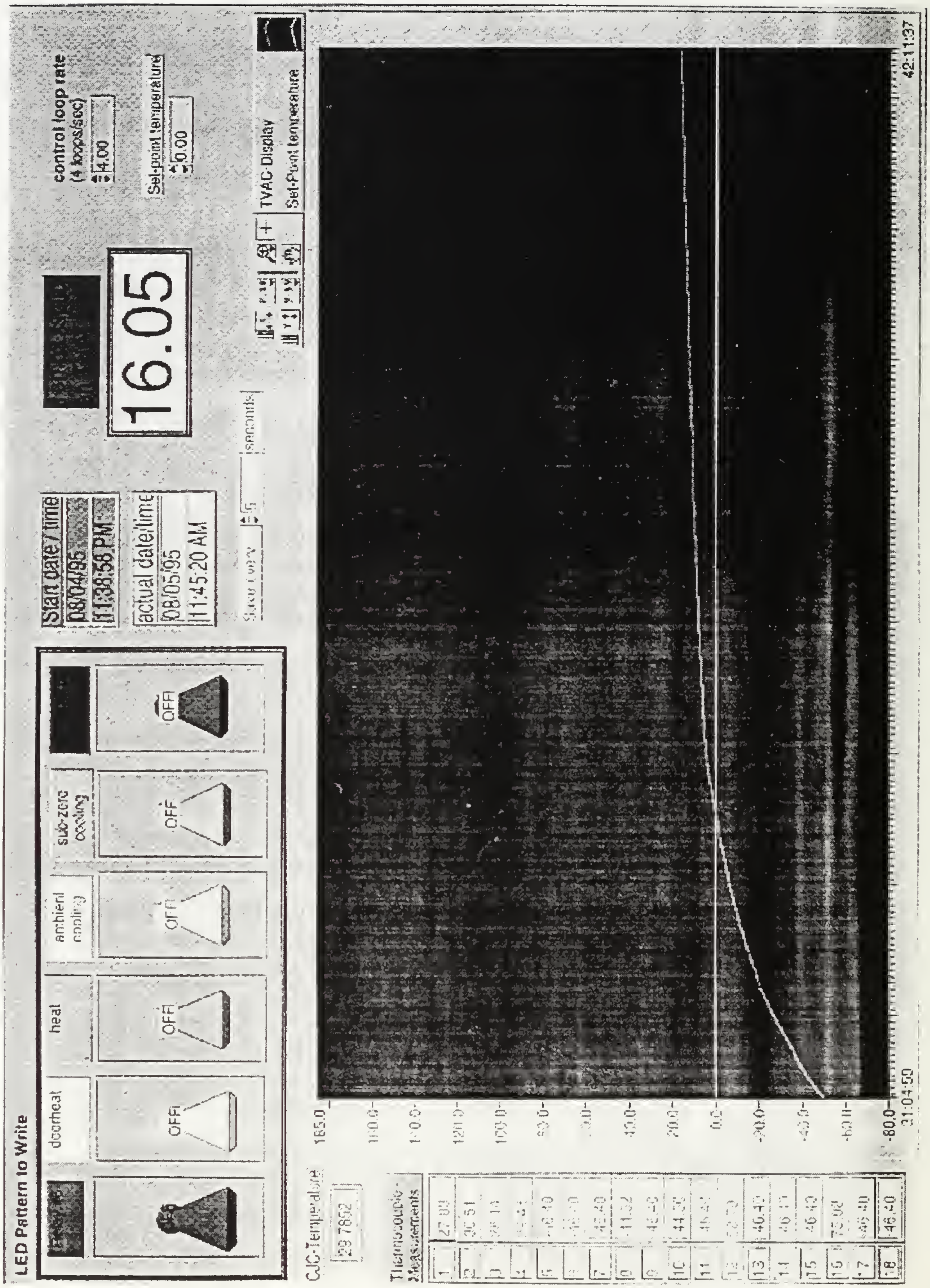


Fig. 50 : "Natural" Reheating

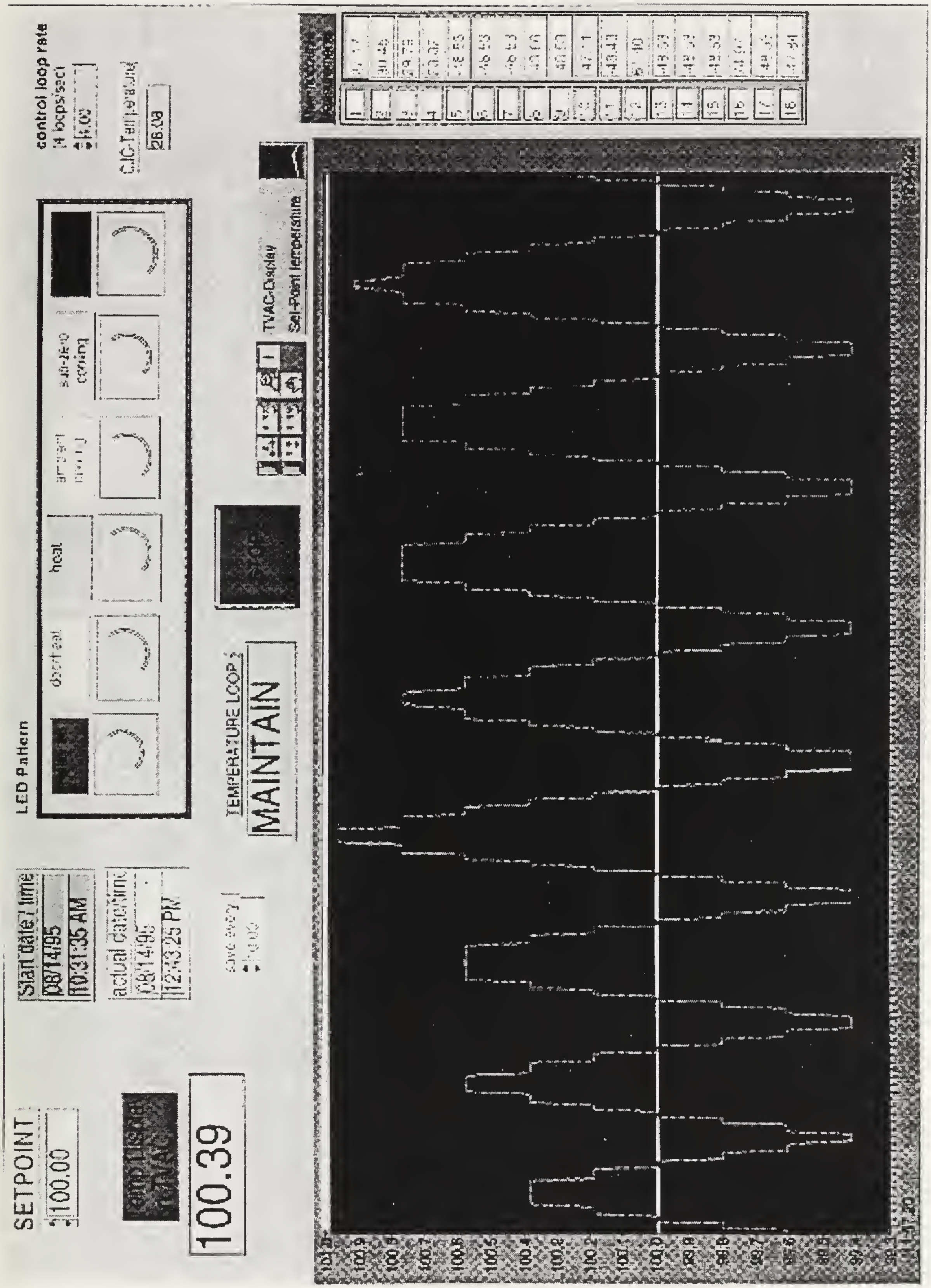


Fig. 51 : *Maintaining around 100 ℃*

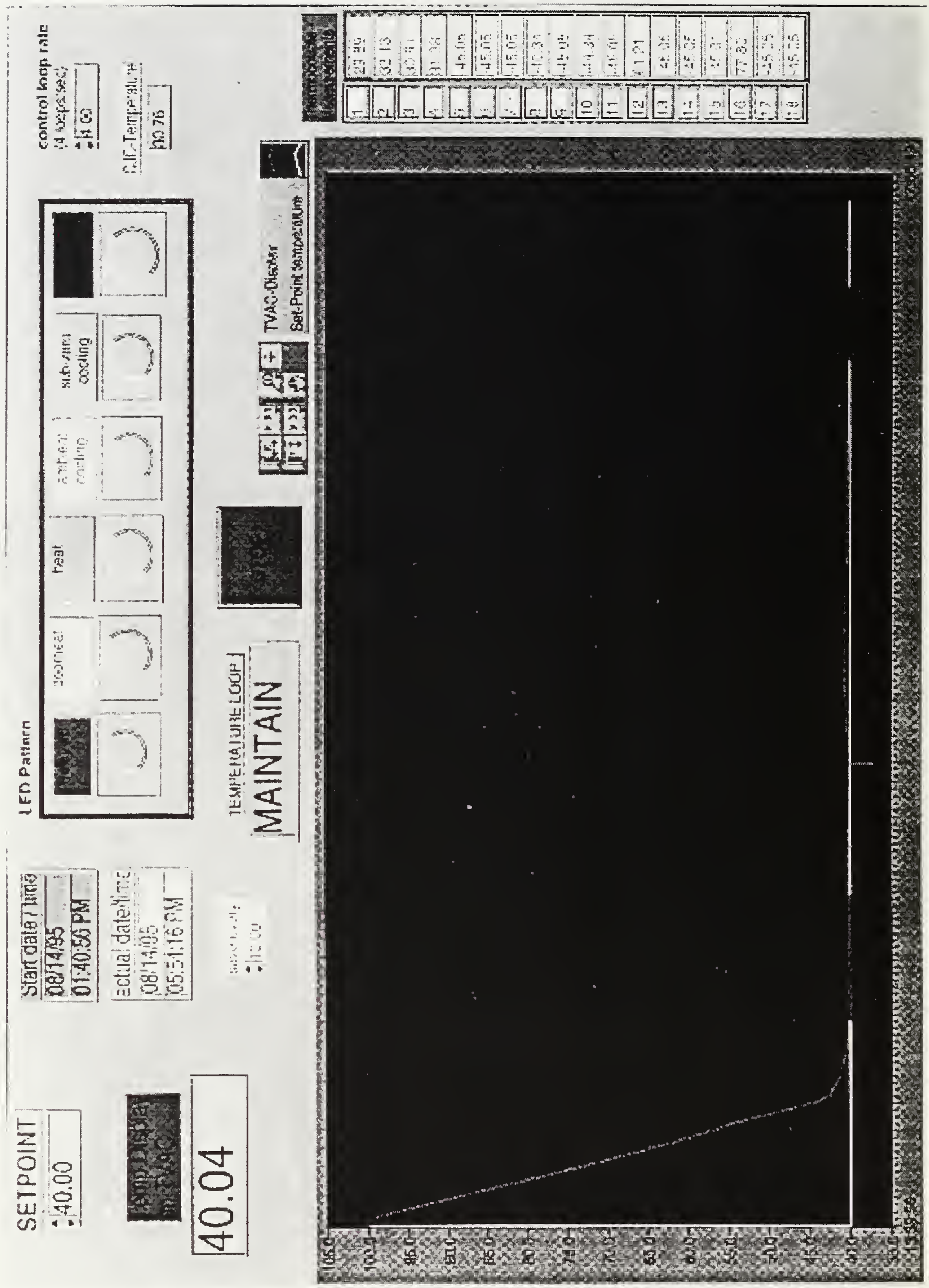


Fig. 52 : *Cooling down to 40℃*

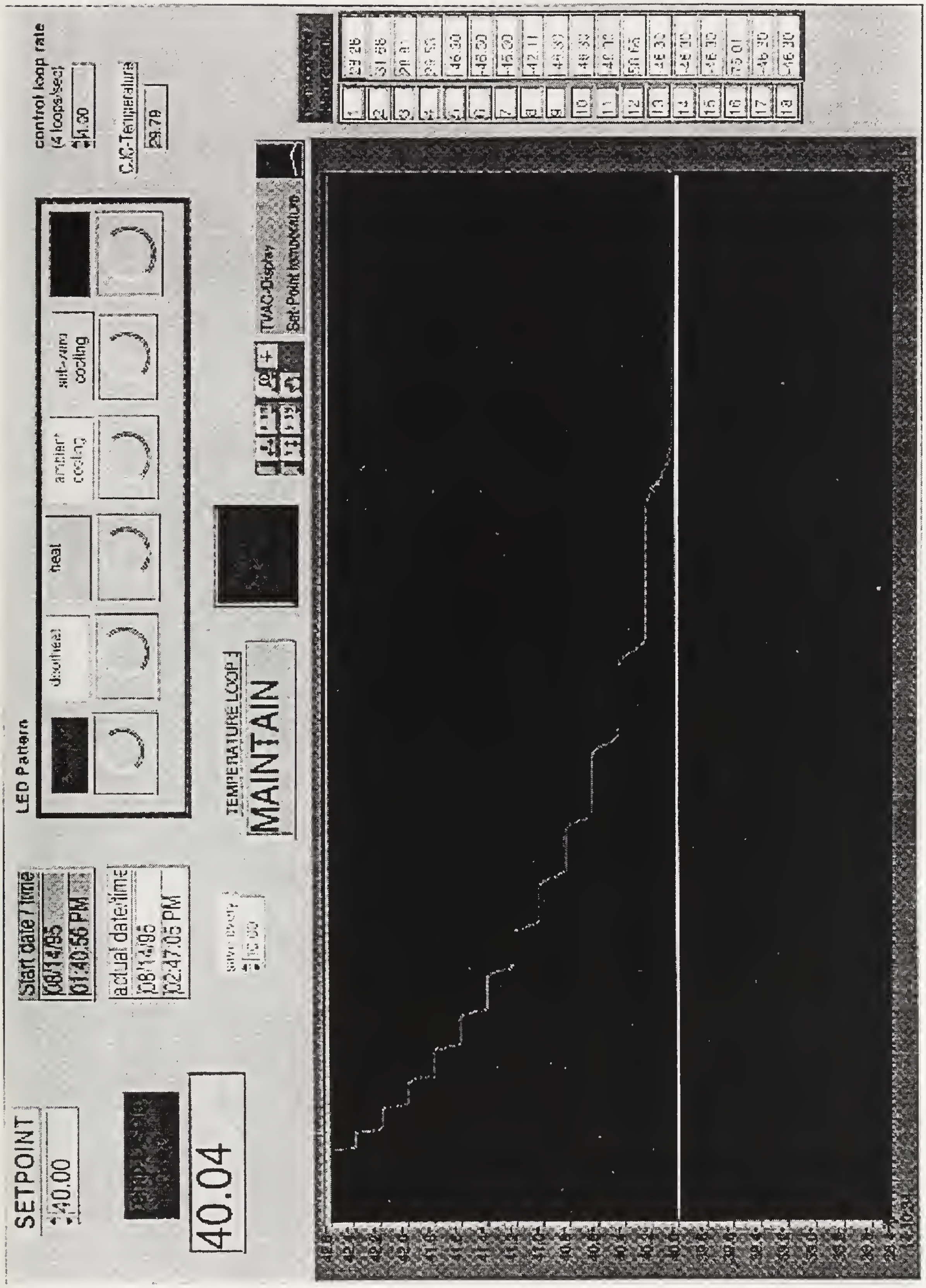


Fig. 53 : Zoom of the approaching temperature curve

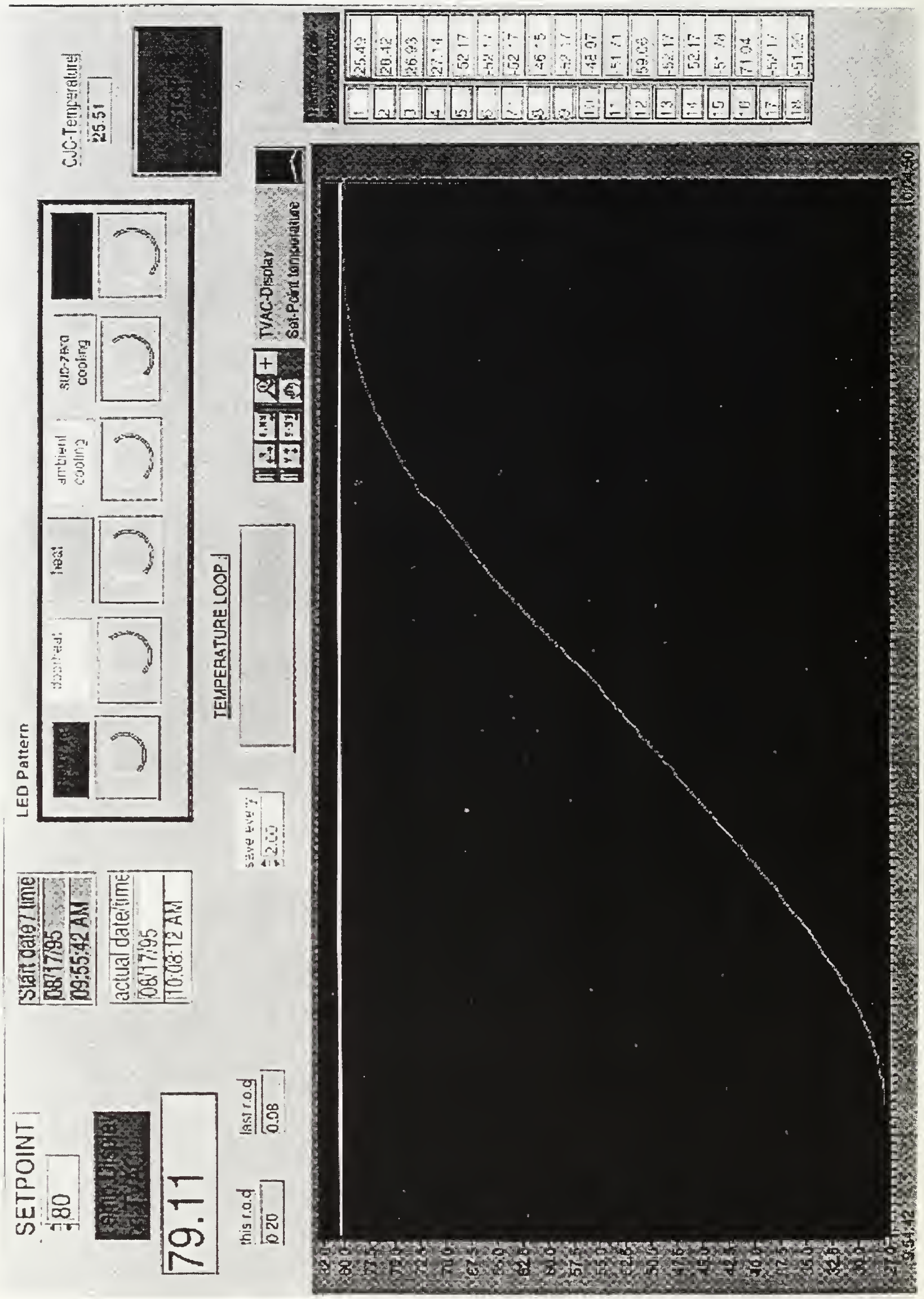


Fig. 54 : Heating from Roomtemperature to 80°C

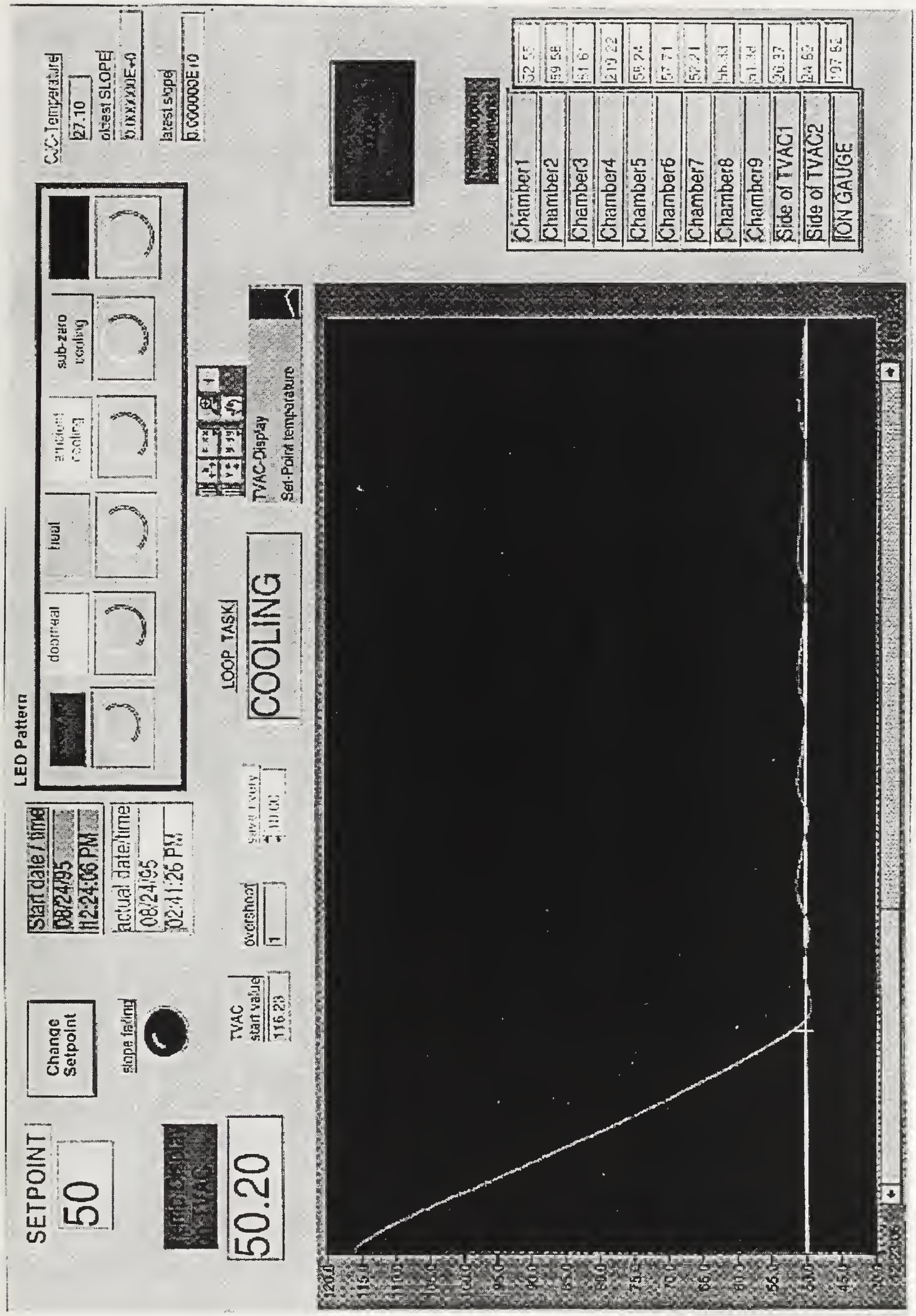


Fig. 55 : *Maintaining at 50℃*

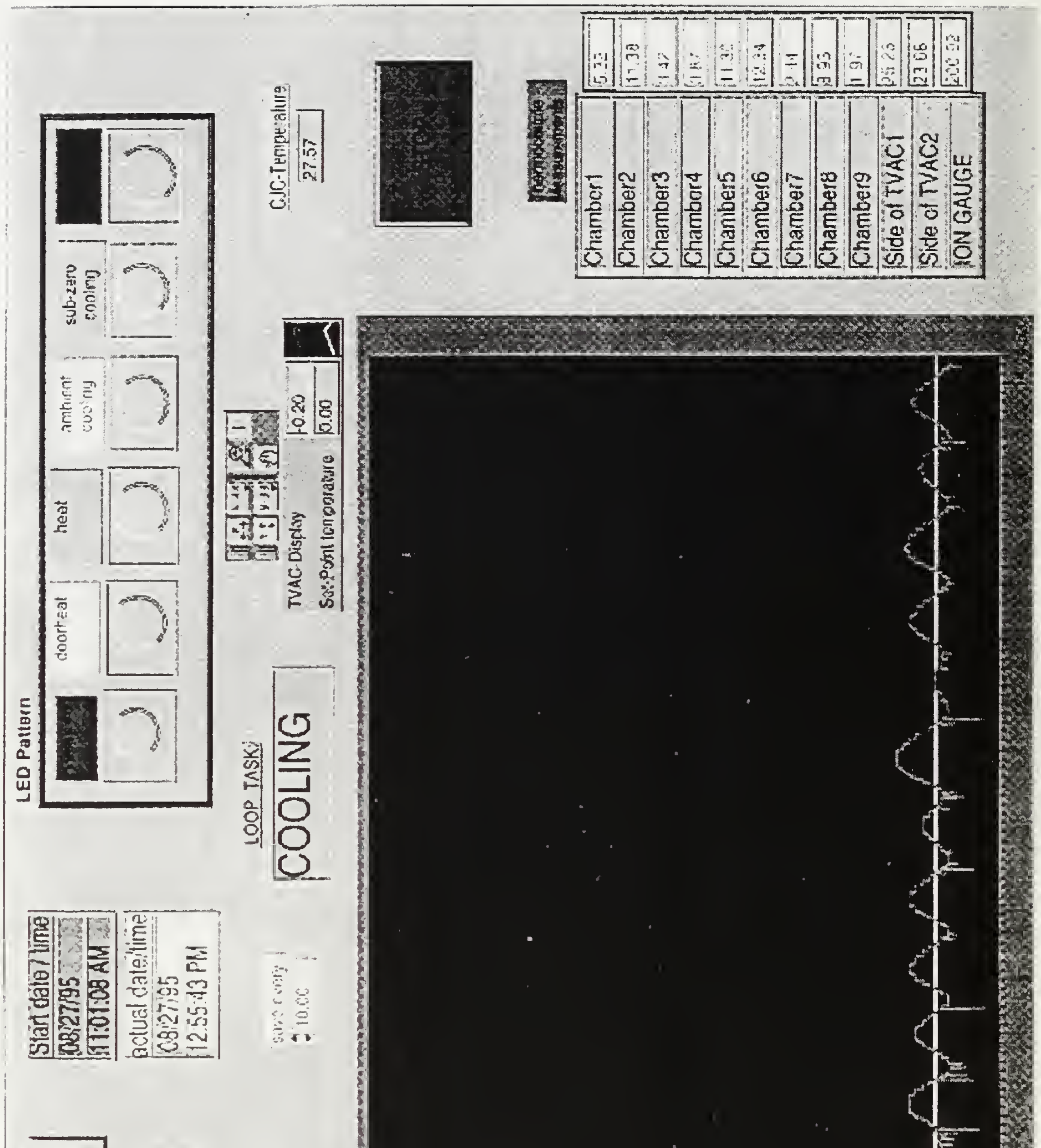


Fig. 56 : Maintaining around 0 ℃

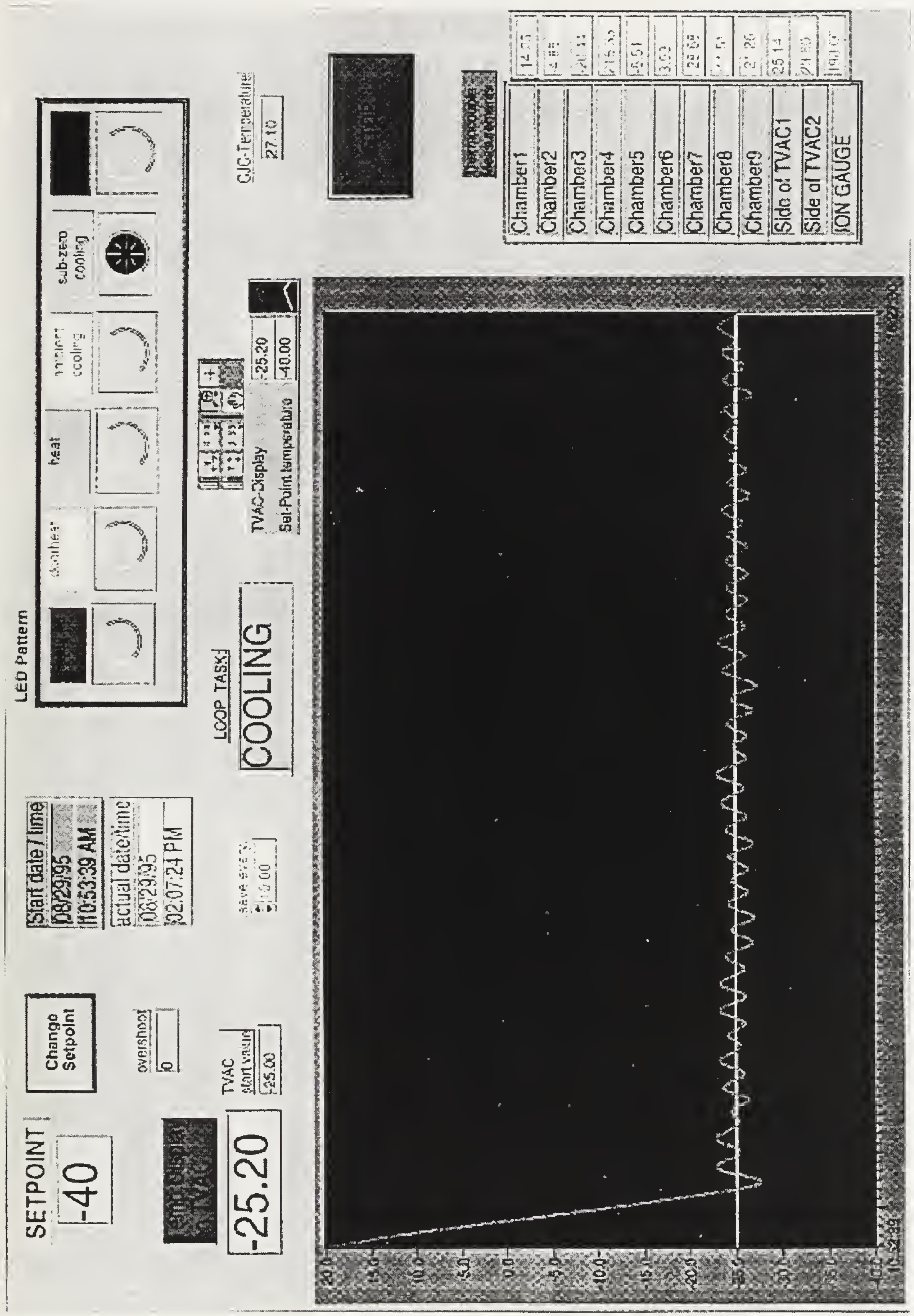


Fig. 57 : Maintaining around -25 ℃

APPENDIX E
TECHNICAL DRAWINGS

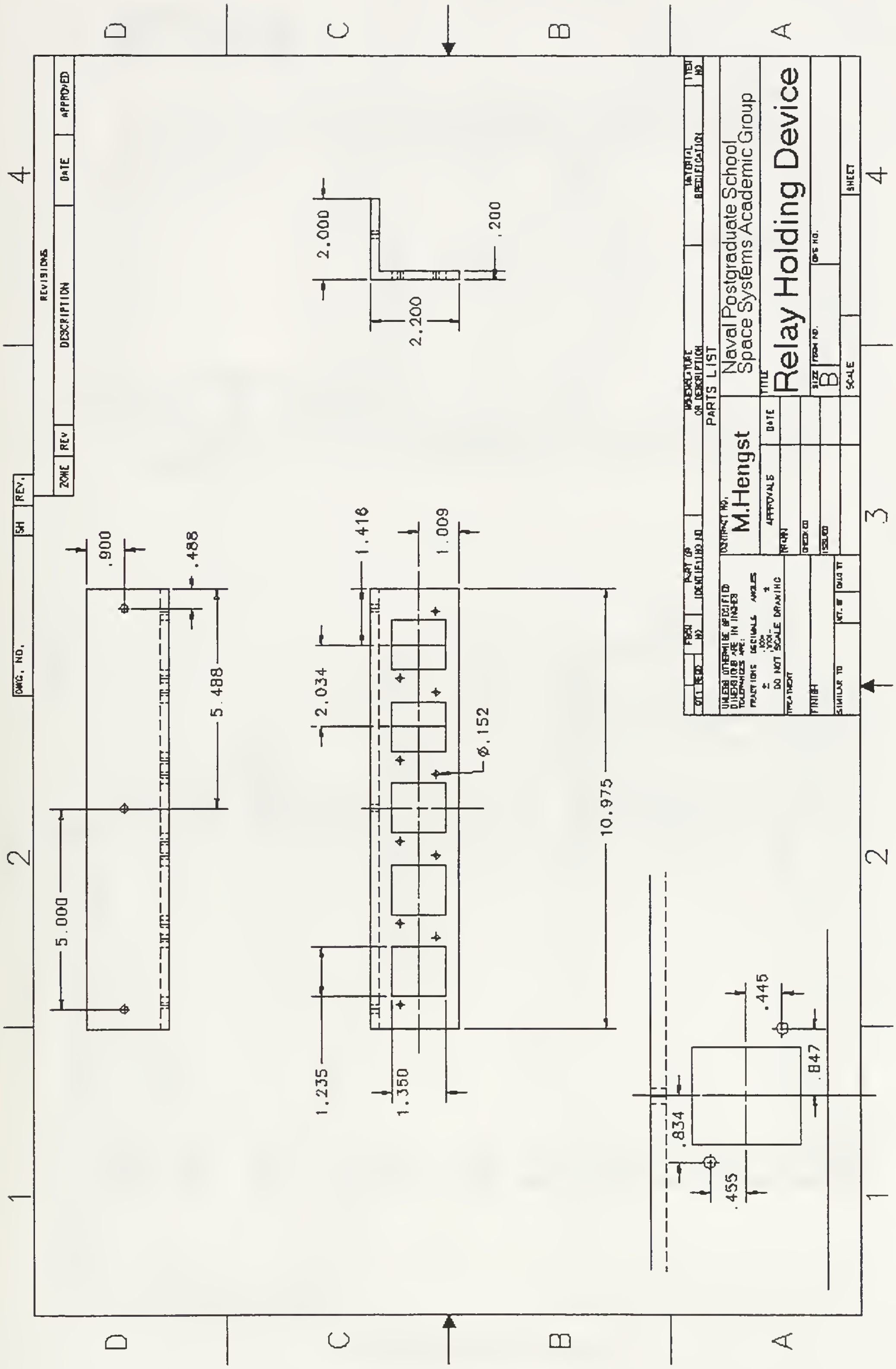


Fig. 58 : Relay Holding Device

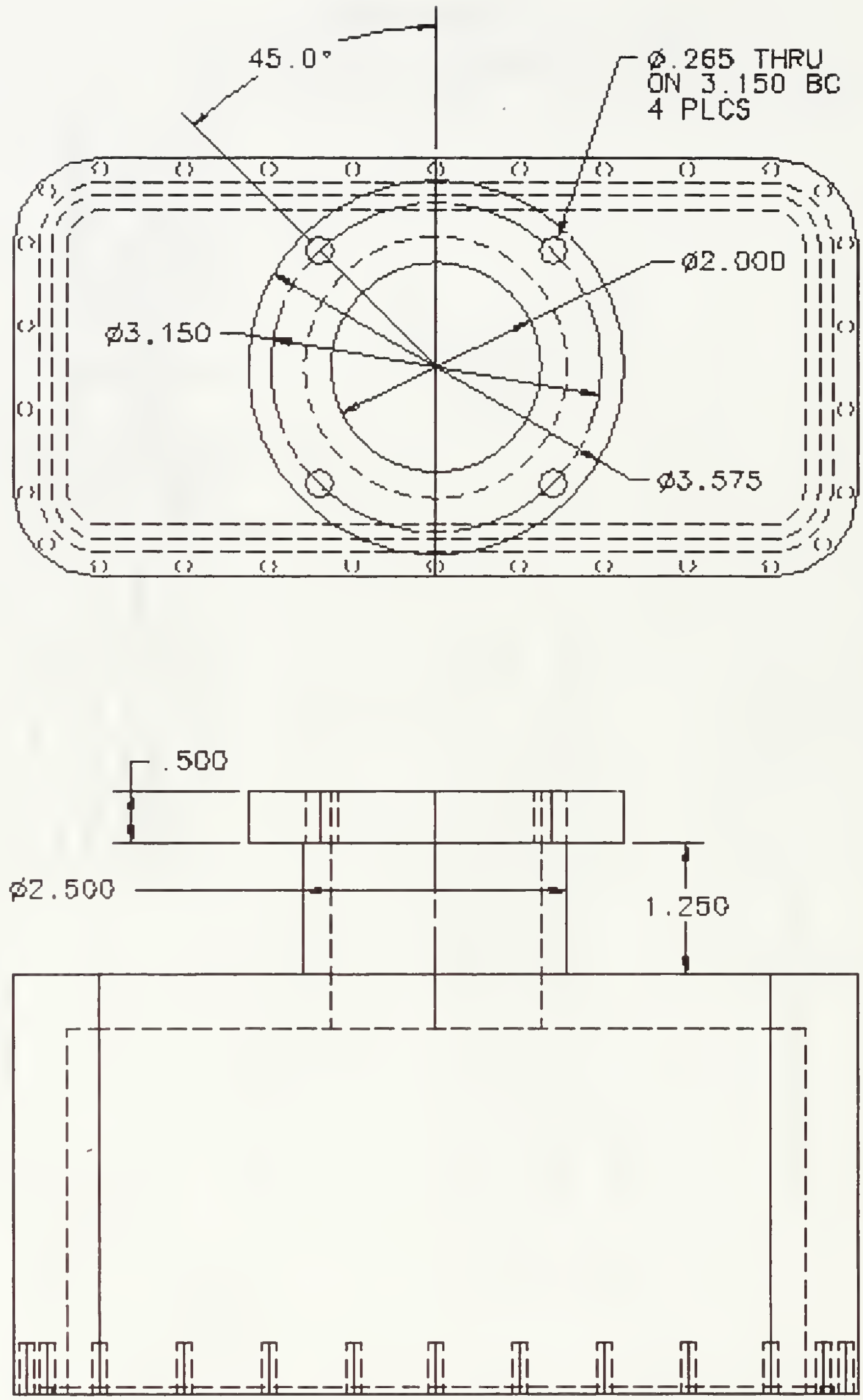


Fig. 59 : Connector Box

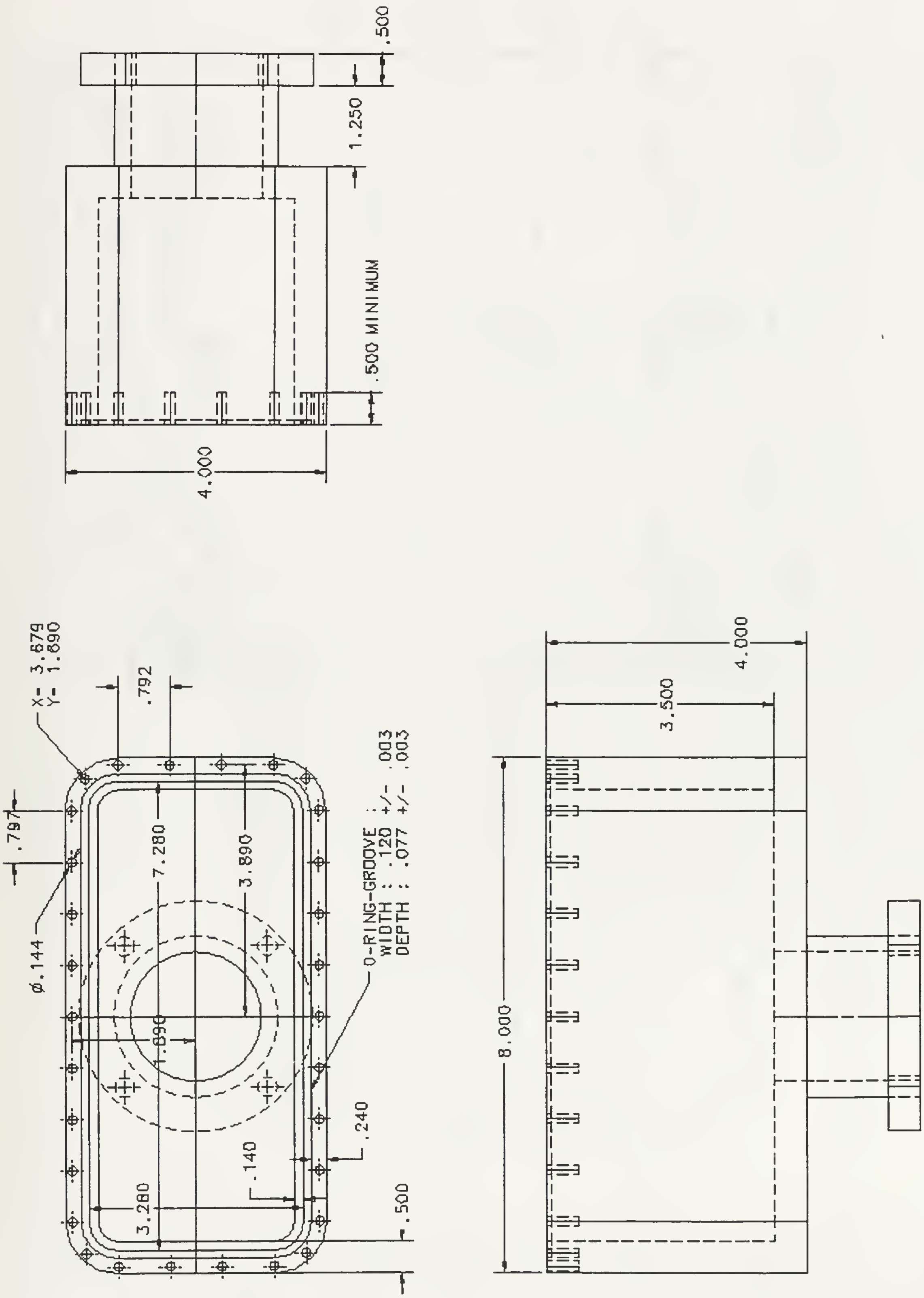


Fig. 60 : *Connector Box*

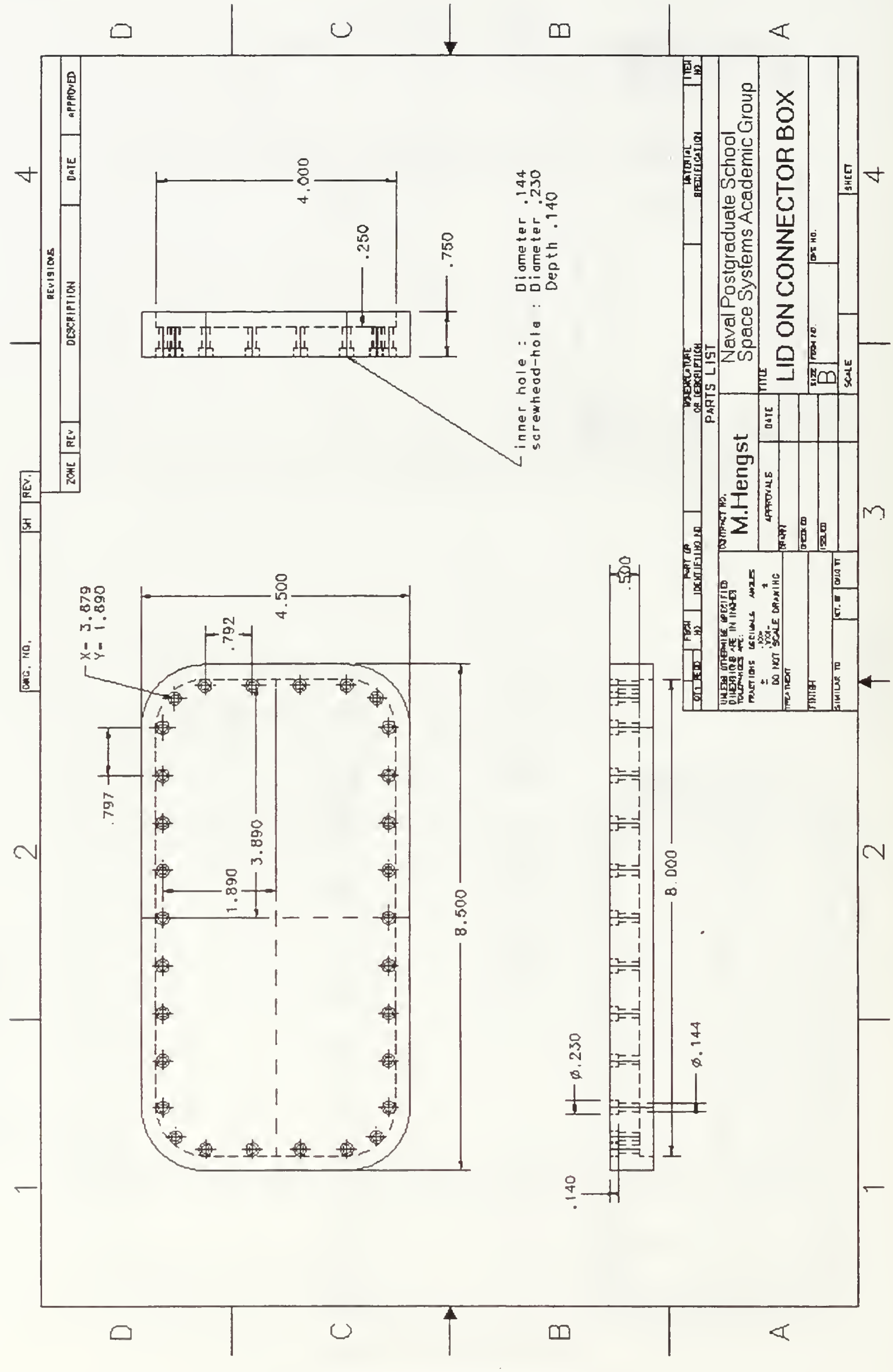


Fig. 61 : Lid on Connector Box

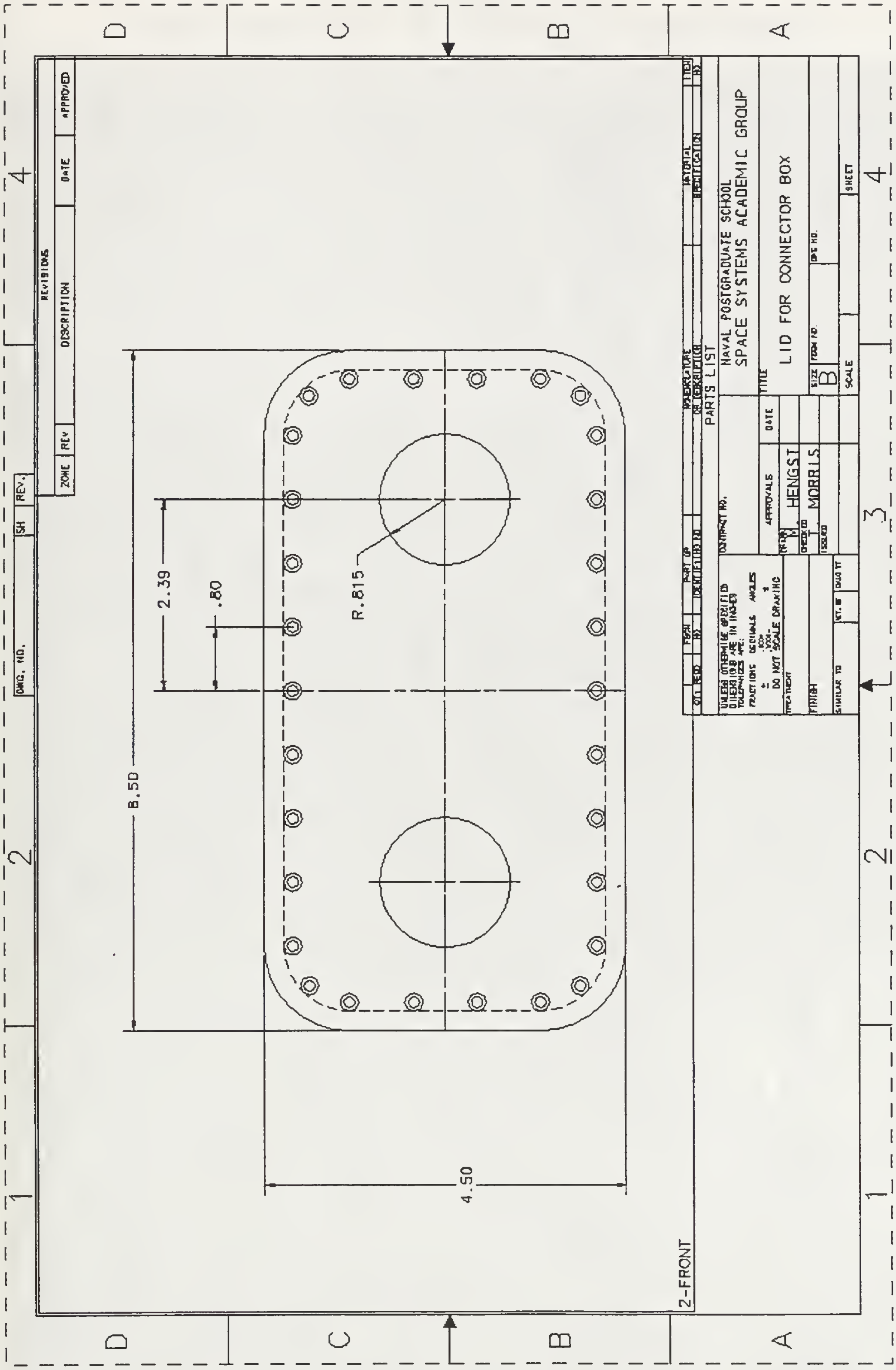
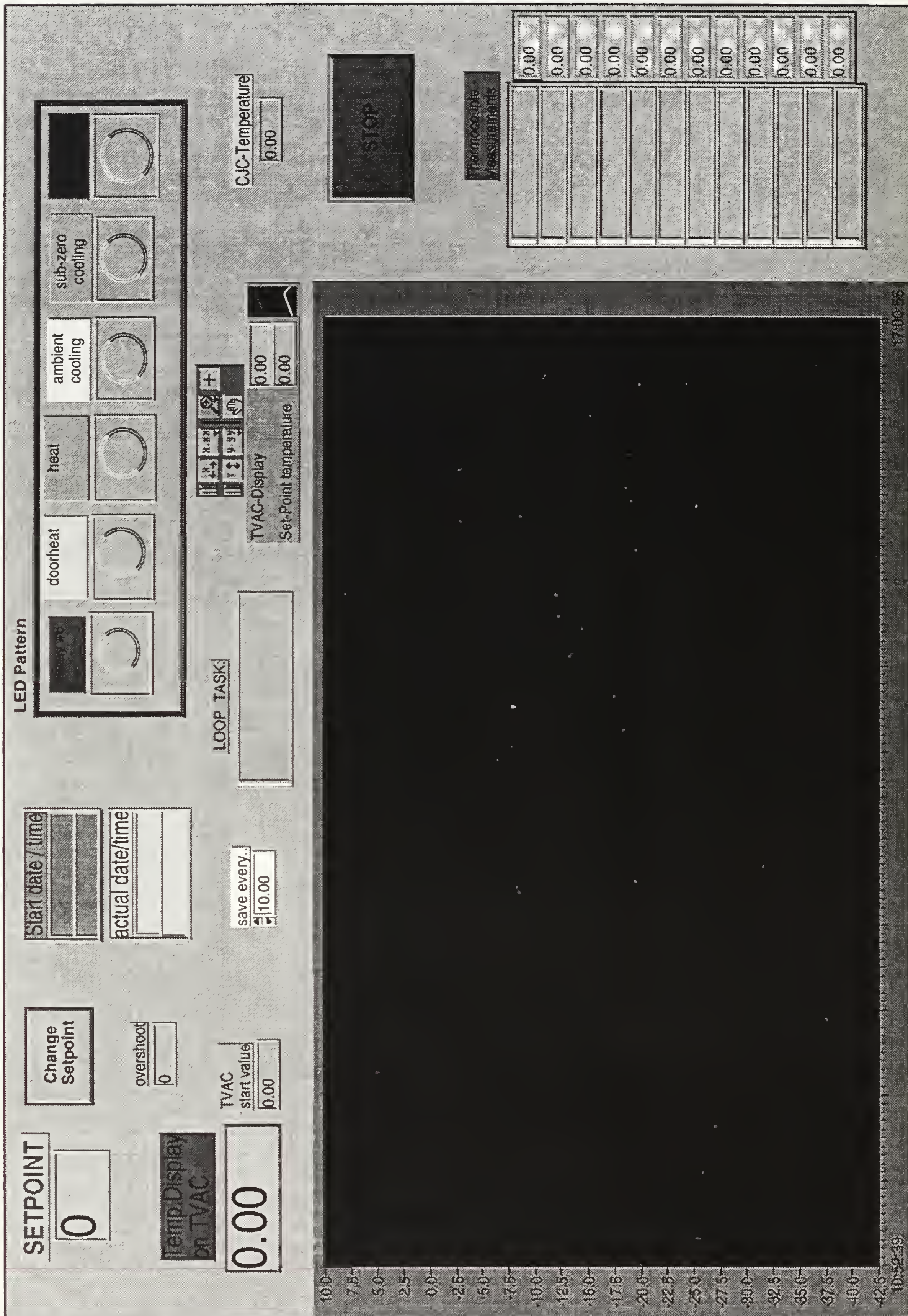


Fig. 62 : Lid with mounting holes

APPENDIX F
DOCUMENTATION OF CONTROL7.VI



Controls and Indicators

SGL **control loop rate (4 loops/sec)**
Enter the scan rate at which you wish to acquire (the number of scans of all the listed channels per second). The scan rate is controlled by LabVIEW " wait until next ms multiple" VI. SInce the scan rate control is in the loop, it can be varied as the application is running. The default is 4 scans/sec.

DBL **save every..**

[abc] **thermocouple channels**
([string]) channels: specifies the list of SCXI analog input channels to scan. The default channel list contains channels 0 through 3 on the module in slot 1 of SCXI chassis 1.

The syntax for SCXI channel strings is:

OBn ! SCx ! MDy ! a:b

n is the onboard data acquisition channel
x is the SCXI chassis ID (from the config utility)
y is the SCXI slot number of the module
a is the first channel of the SCXI channel range
b is the ending channel in the SCXI channel range

For one chassis applications, n = 0. If you have more than one chassis daisy-chained to the same data acquisition board, n = 1 for the second chassis in the chain, etc.

If you only want to acquire from one SCXI channel, omit the ":b" from the end of the string.

When you are scanning multiplexed SCXI modules, you must scan channels in ascending consecutive order; that is, b must be greater than a.

To scan multiple modules, enter an SCXI string like the one above for each module in the channels array; one element in the array for each module to be scanned.

You can also enter strings for modules in different chassis to be scanned at the same time. Remember to use the correct n and x values.

abc **channel (0)**
([string]) channels: specifies the list of SCXI analog input channels to scan. The default channel list contains channels 0 through 3 on the module in slot 1 of SCXI chassis 1.

The syntax for SCXI channel strings is:

OBn ! SCx ! MDy ! a:b

n is the onboard data acquisition channel
x is the SCXI chassis ID (from the config utility)
y is the SCXI slot number of the module
a is the first channel of the SCXI channel range
b is the ending channel in the SCXI channel range

For one chassis applications, n = 0. If you have more than one chassis daisy-chained to the same data acquisition board, n = 1 for the second chassis in the chain, etc.

If you only want to acquire from one SCXI channel, omit the ":b" from the end of the string.

When you are scanning multiplexed SCXI modules, you must scan channels in ascending consecutive order; that is, b must be greater than a.

To scan multiple modules, enter an SCXI string like the one above for each module in the channels array; one element in the array for each module to be scanned.

You can also enter strings for modules in different chassis to be scanned at the same time. Remember to use the correct n and x values.

U16 **TC type**
Type of thermocouple wire used.

TF **stop**

[abc] **TC names array**

[abc]

[TF] **CHANGE SETPOINT ?**

[DBL] **Temp.Display on TVAC**

[abc]

[abc] **actual date/time**

[abc] **Start date / time**

[abc]

[DBL] **time at last save**

[DBL] **time now**

[DBL] **SETPOINT**

[abc] **Thermocouple - Measurements**

[abc]

[DBL] **CJC-Temperature**

LED Pattern
When writing to a port use the LED controls within this cluster to control the states of the lines in the port. When an LED is illuminated the corresponding bit of the port is set to high. When the LED is off the bit is set

[TF] **0**

[TF] **1**

[TF] **2**

[TF] **3**

[TF] **4**

[TF] **Relay #6**

Strip Chart
Displays the data. The delta X is determined by the initial reading of Delay. Any subsequent changes in Delay do not affect the X axis until the next run of the VI.

[DBL] **Thermocouple-Values**

[DBL]

[abc] **LOOP TASK:**

[EXT] **latest slope**

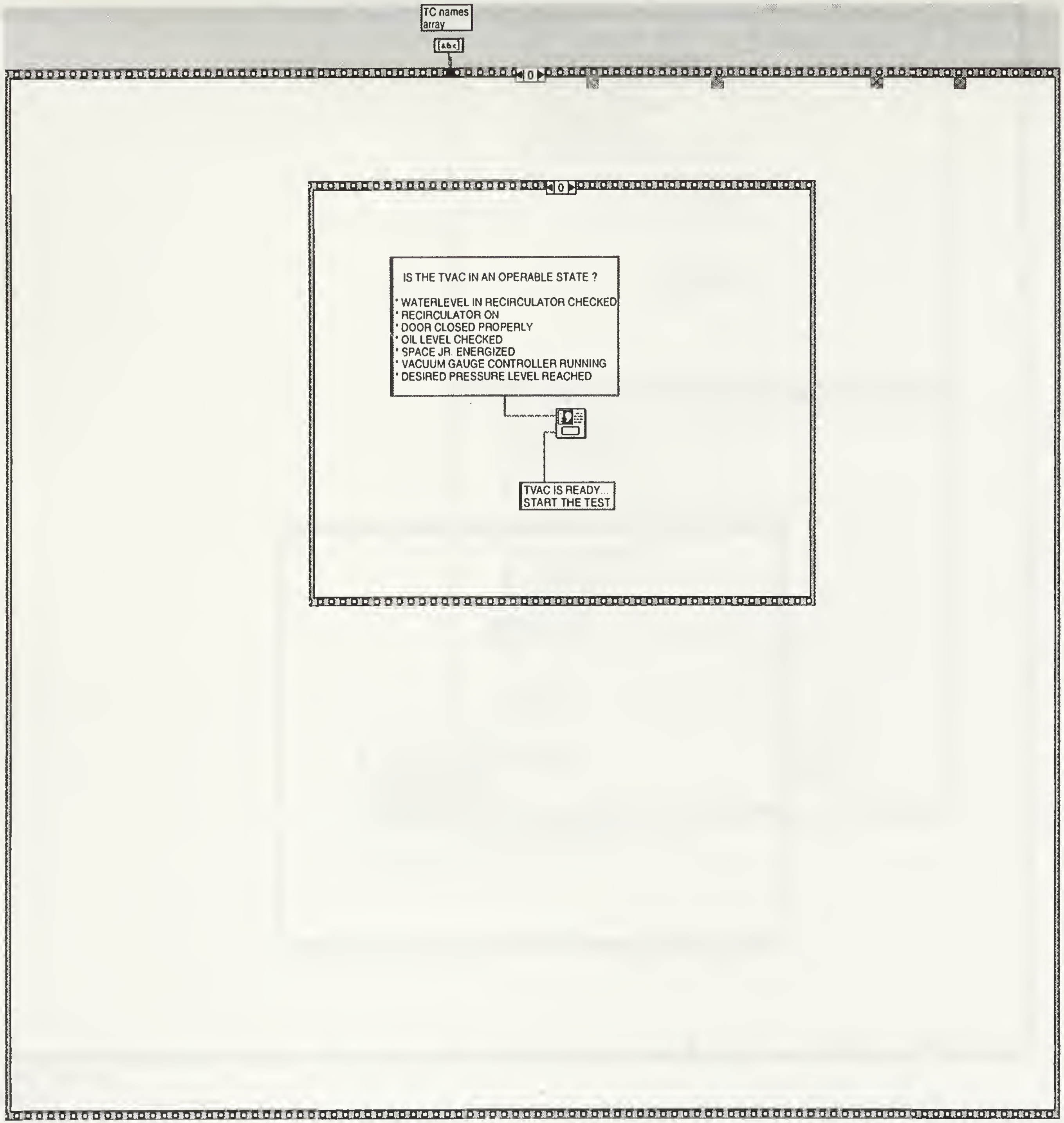
[EXT] **oldest SLOPE**

[DBL] **TVAC start value**

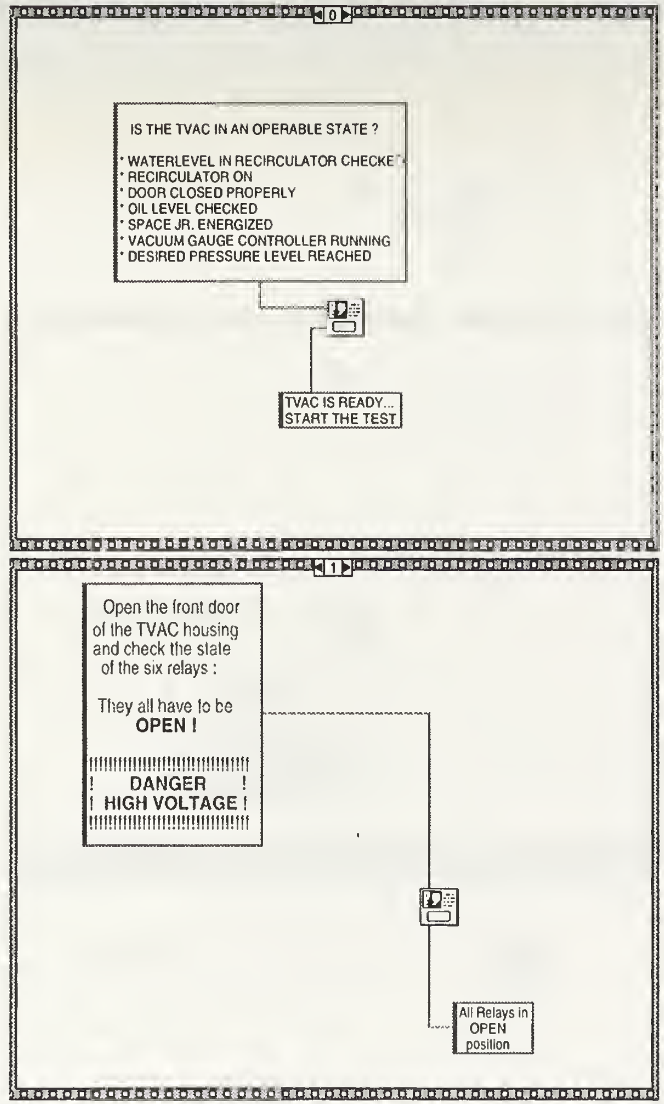
[DBL] **overshoot**

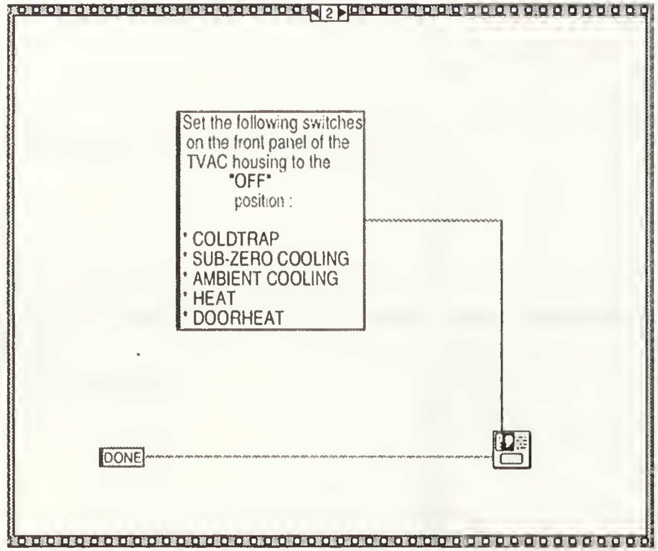
[TF] **slope falling**

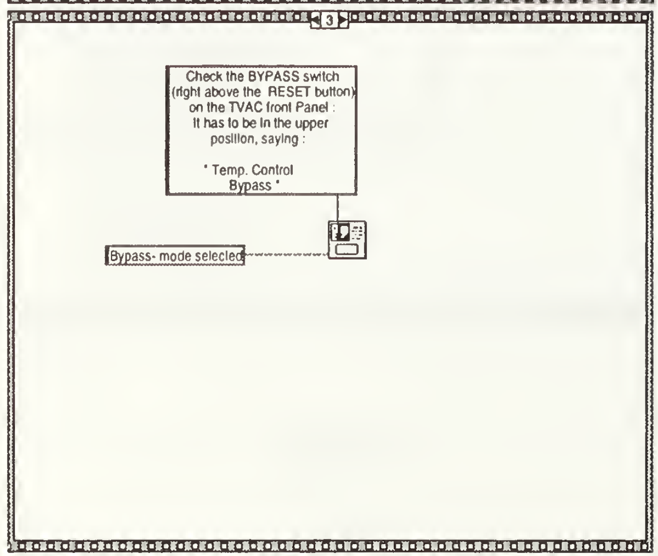
Block Diagram



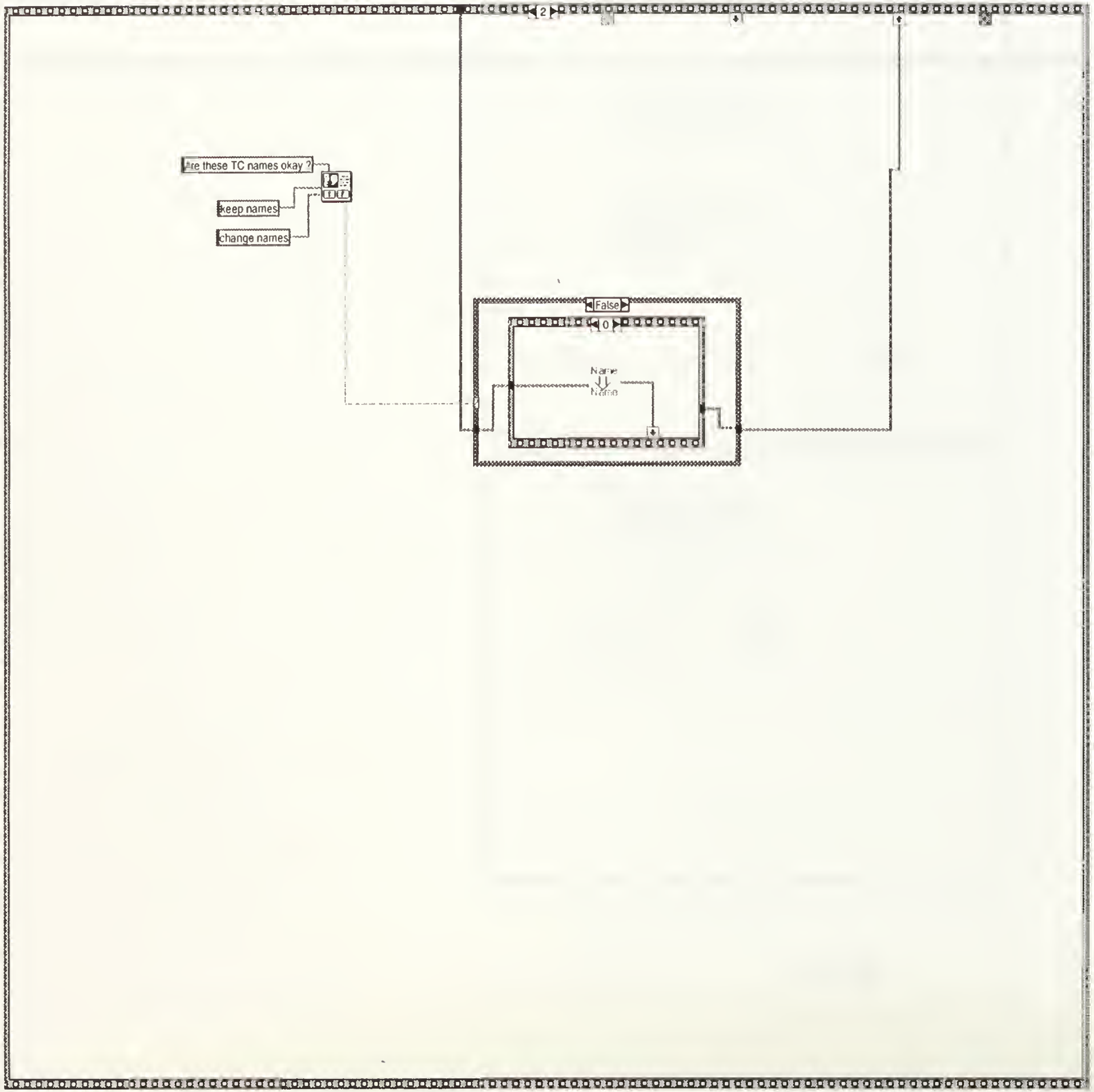

0
0
IS THE TVAC IN AN OPERABLE STATE ?
* WATERLEVEL IN RECIRCULATOR CHECKED
* RECIRCULATOR ON
* DOOR CLOSED PROPERLY
* OIL LEVEL CHECKED
* SPACE JR. ENERGIZED
* VACUUM GAUGE CONTROLLER RUNNING
* DESIRED PRESSURE LEVEL REACHED
TVAC IS READY...
START THE TEST

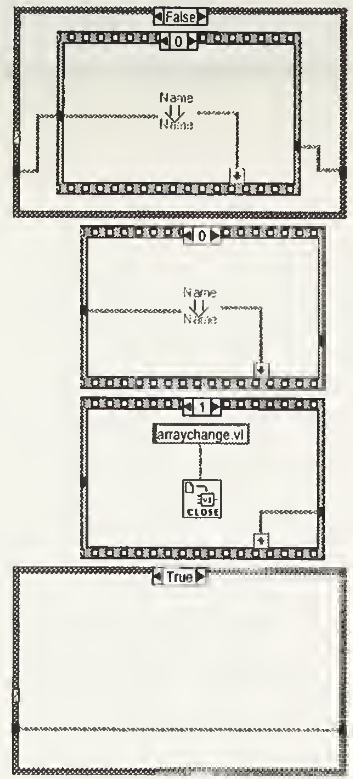

0
IS THE TVAC IN AN OPERABLE STATE ?
* WATERLEVEL IN RECIRCULATOR CHECKE
* RECIRCULATOR ON
* DOOR CLOSED PROPERLY
* OIL LEVEL CHECKED
* SPACE JR. ENERGIZED
* VACUUM GAUGE CONTROLLER RUNNING
* DESIRED PRESSURE LEVEL REACHED
TVAC IS READY...
START THE TEST
1
Open the front door
of the TVAC housing
and check the state
of the six relays :
They all have to be
OPEN !
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! DANGER !
! HIGH VOLTAGE !
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
All Relays in
OPEN
position

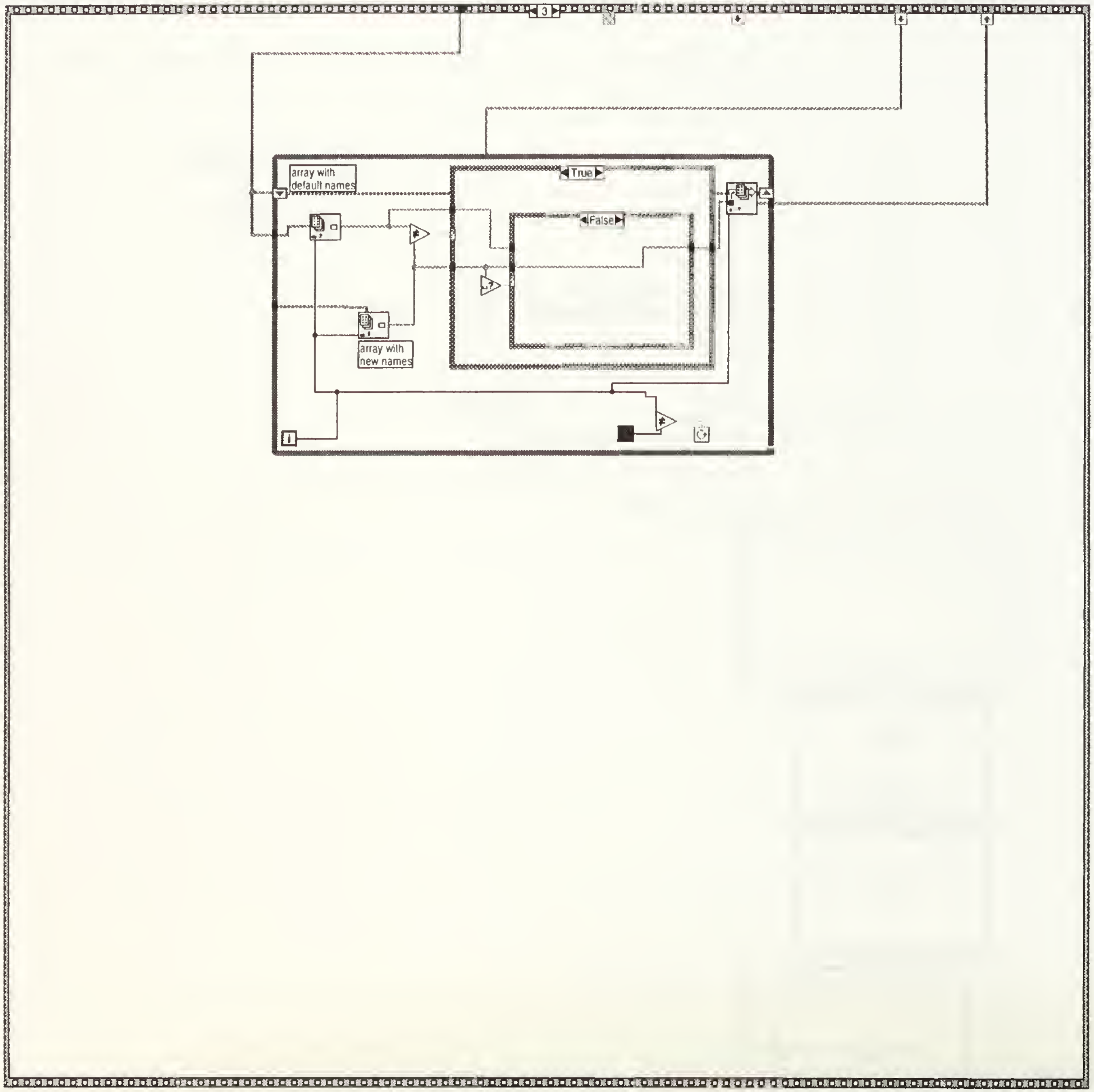

2
Set the following switches on the front panel of the TVAC housing to the
"OFF"
position :
* COLDTRAP
* SUB-ZERO COOLING
* AMBIENT COOLING
* HEAT
* DOORHEAT
DONE

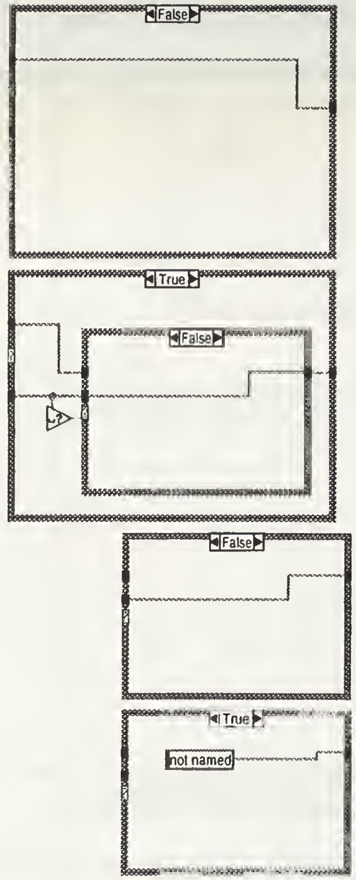

3
Check the BYPASS switch (right above the RESET button) on the TVAC front Panel :
It has to be in the upper position, saying :
" Temp. Control Bypass "
Bypass- mode selected

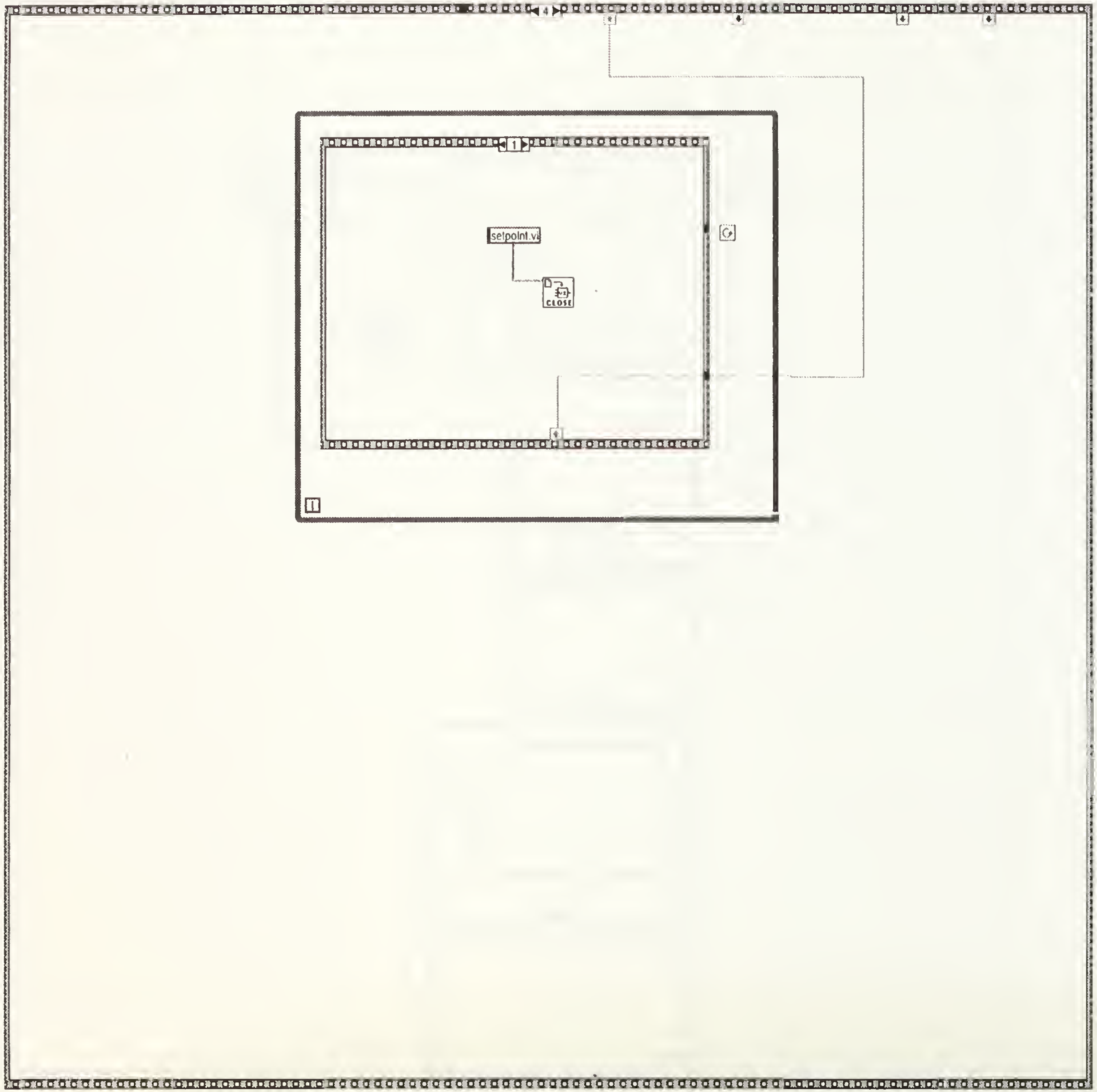

1

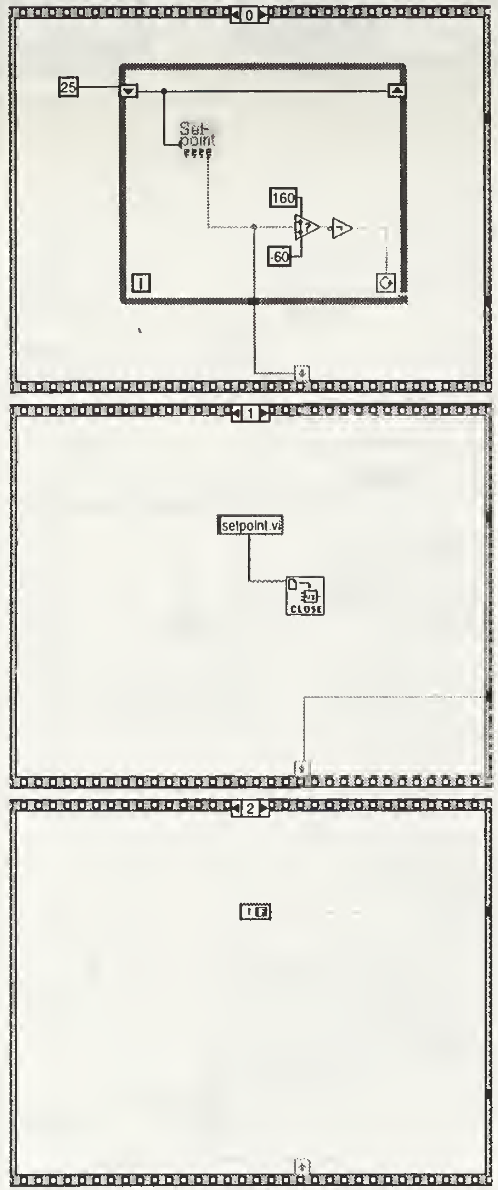

2
Are these TC names okay ?
keep names
change names
False
0
Name
Name

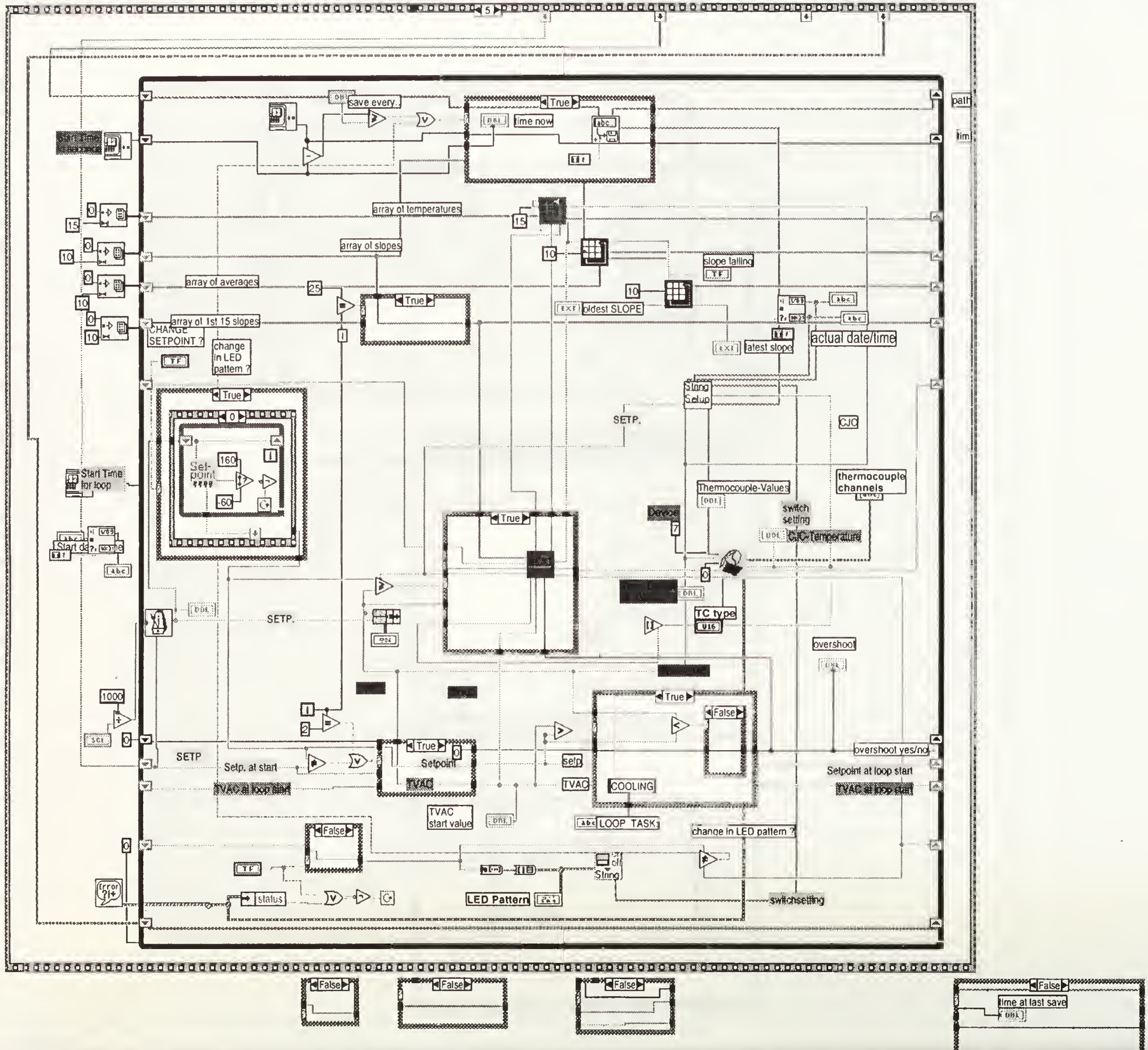

False
0
Name
Name
0
Name
Name
1
arraychange.vi
CLOSE
True


3
True
False
array with
default names
array with
new names

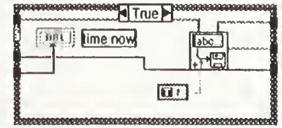

False
True
False
False
True
not named


4
1
setpoint.vi
CLOSE

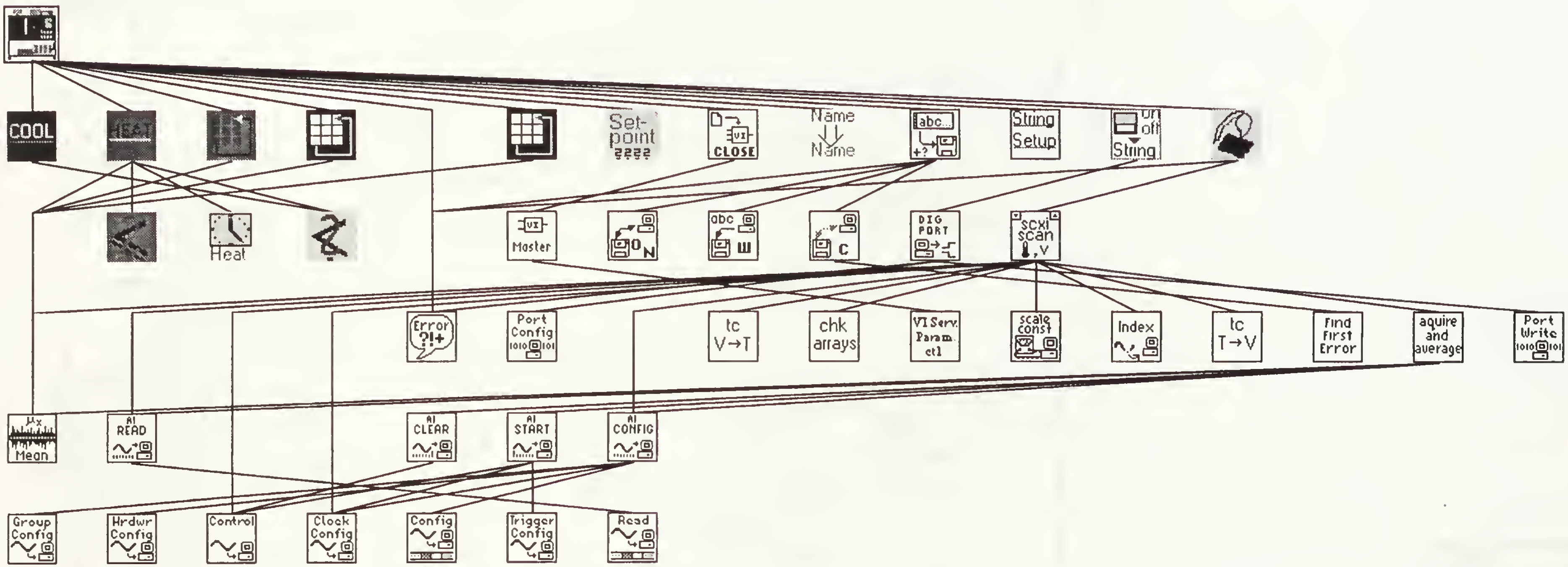

0
25
Set point
160
-60
1
setpoint.vi
CLOSE
2


5
save every
True
time now
array of temperatures
15
array of slopes
10
slope falling
array of averages
25
True
oldest SLOPE
array of 1st 15 slopes
latest slope
actual date/time
CHANGE SETPOINT ?
change in LED pattern ?
True
0
160
Set-point
-60
String Setup
SETP.
CJC
Start Time for loop
Thermocouple-Values
thermocouple channels
switch setting
CJC-Temperature
7
True
SETP.
TC type
overshoot
1000
SETP
Setp. at start
True
Setpoint
TVAC
setp
TVAC
COOLING
overshoot yes/no
Setpoint at loop start
TVAC start value
LOOP TASK
change in LED pattern ?
False
status
LED Pattern
String
switchsetting
path
False
False
False
False
time at last save


False
True
0
True
True
Setpoint
0
EVAC
False
COOL
True
False
True
1
HEATING
False
True
1
True
False
COOLING
False
True
1
True
0
160
Set-
point
-60
0
160
Set-
point
-60
1
setpoint.vi
CLOSE



Position in Hierarchy



List of SubVIs

reading thermocouples2.subvi
F:\LABVIEW\VIS\TVAC2.LLB\reading thermocouples2.subvi

switching2.vi
F:\LABVIEW\VIS\TVAC2.LLB\switching2.vi

String setup.vi
F:\LABVIEW\VIS\TVAC2.LLB\String setup.vi

Write Characters To File.vi
C:\LABVIEW\vi.lib\UTILITY\FILE.LLB\Write Characters To File.vi

General Error Handler.vi
C:\LABVIEW\vi.lib\UTILITY\ERROR.LLB\General Error Handler.vi

arraychange.vi
C:\LABVIEW\EXAMPLES\DAQ\TVAC2.LLB\arraychange.vi

Close Panel.vi
C:\LABVIEW\vi.lib\UTILITY\VICTL.LLB\Close Panel.vi

Setpoint.vi
F:\LABVIEW\VIS\TVAC2.LLB\Setpoint.vi

array8c.vi
C:\LABVIEW\EXAMPLES\DAQ\TVAC2.LLB\array8c.vi

array8b.vi
C:\LABVIEW\EXAMPLES\DAQ\TVAC2.LLB\array8b.vi

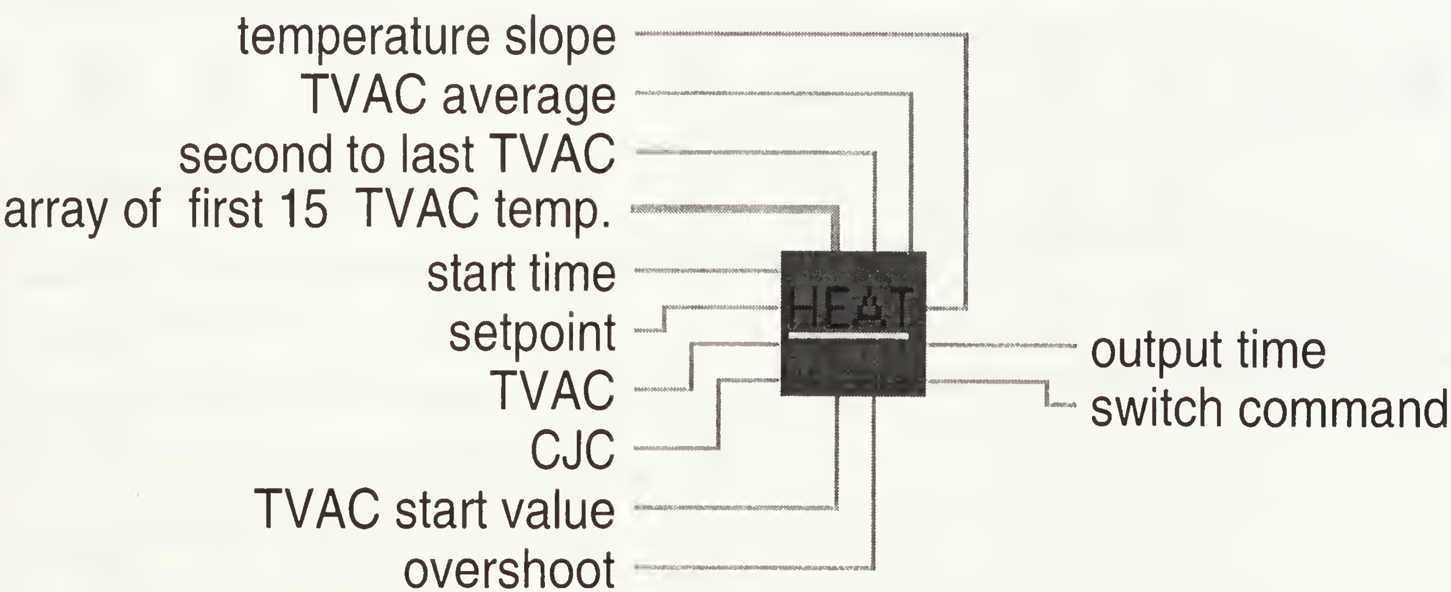
array8a.vi
C:\LABVIEW\EXAMPLES\DAQ\TVAC2.LLB\array8a.vi

Heating.vi
C:\LABVIEW\EXAMPLES\DAQ\TVACUUM.LLB\Heating.vi

Cooling.vi
C:\LABVIEW\EXAMPLES\DAQ\TVACUUM.LLB\Cooling.vi

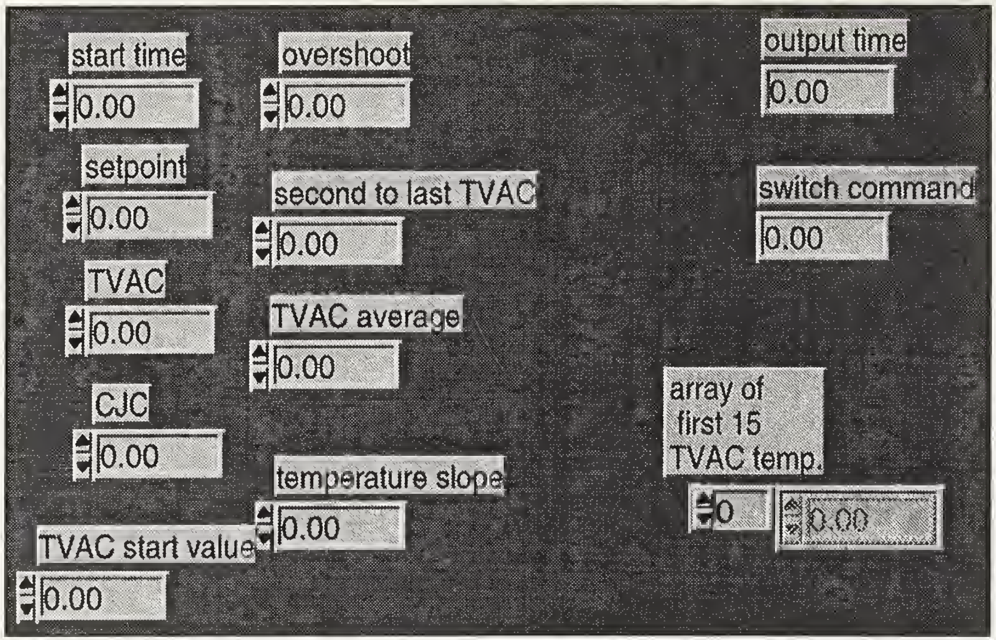
APPENDIX G
DOCUMENTATION OF HEATING.VI

Connector Pane



Heating.vi

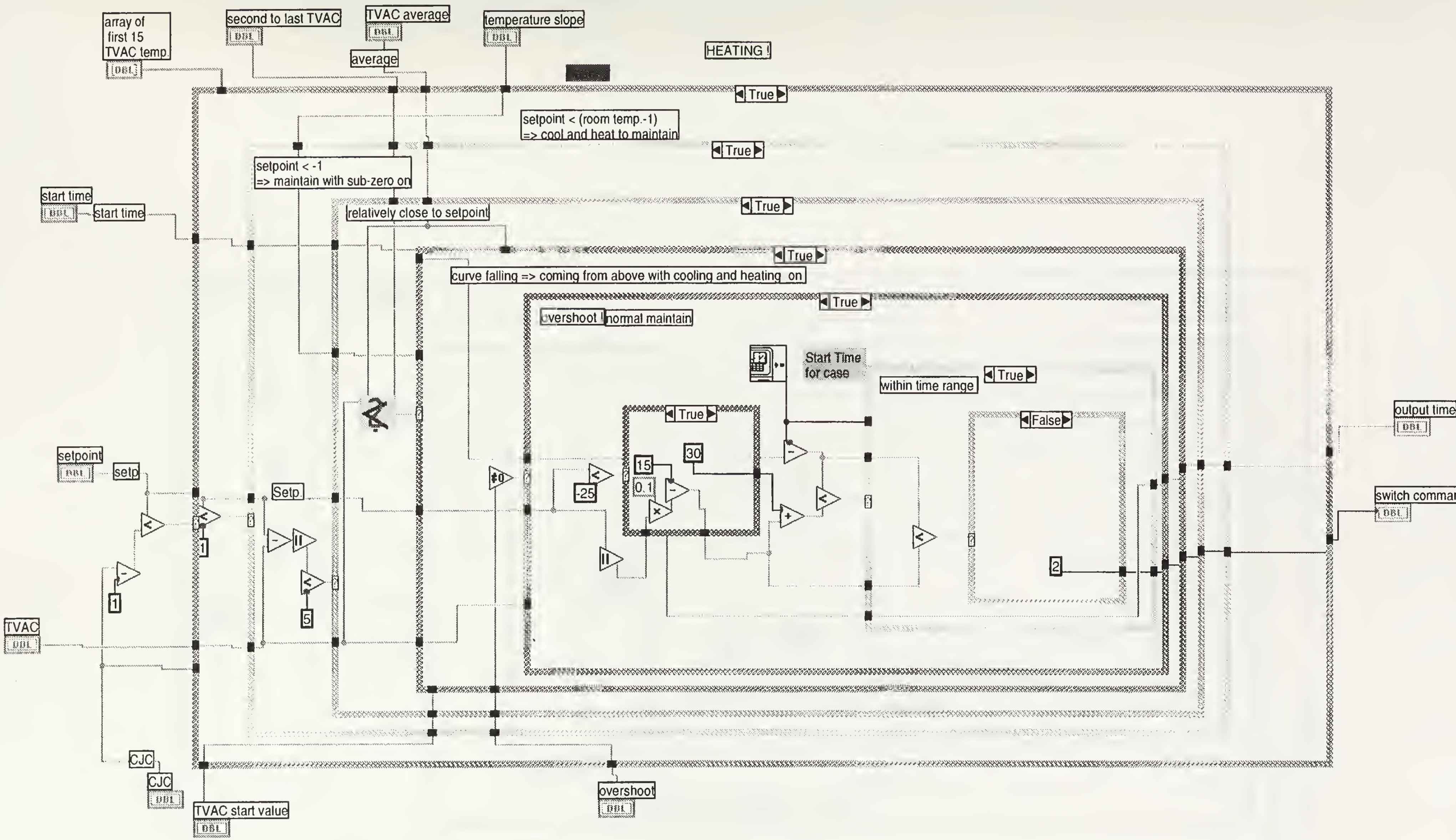
Front Panel

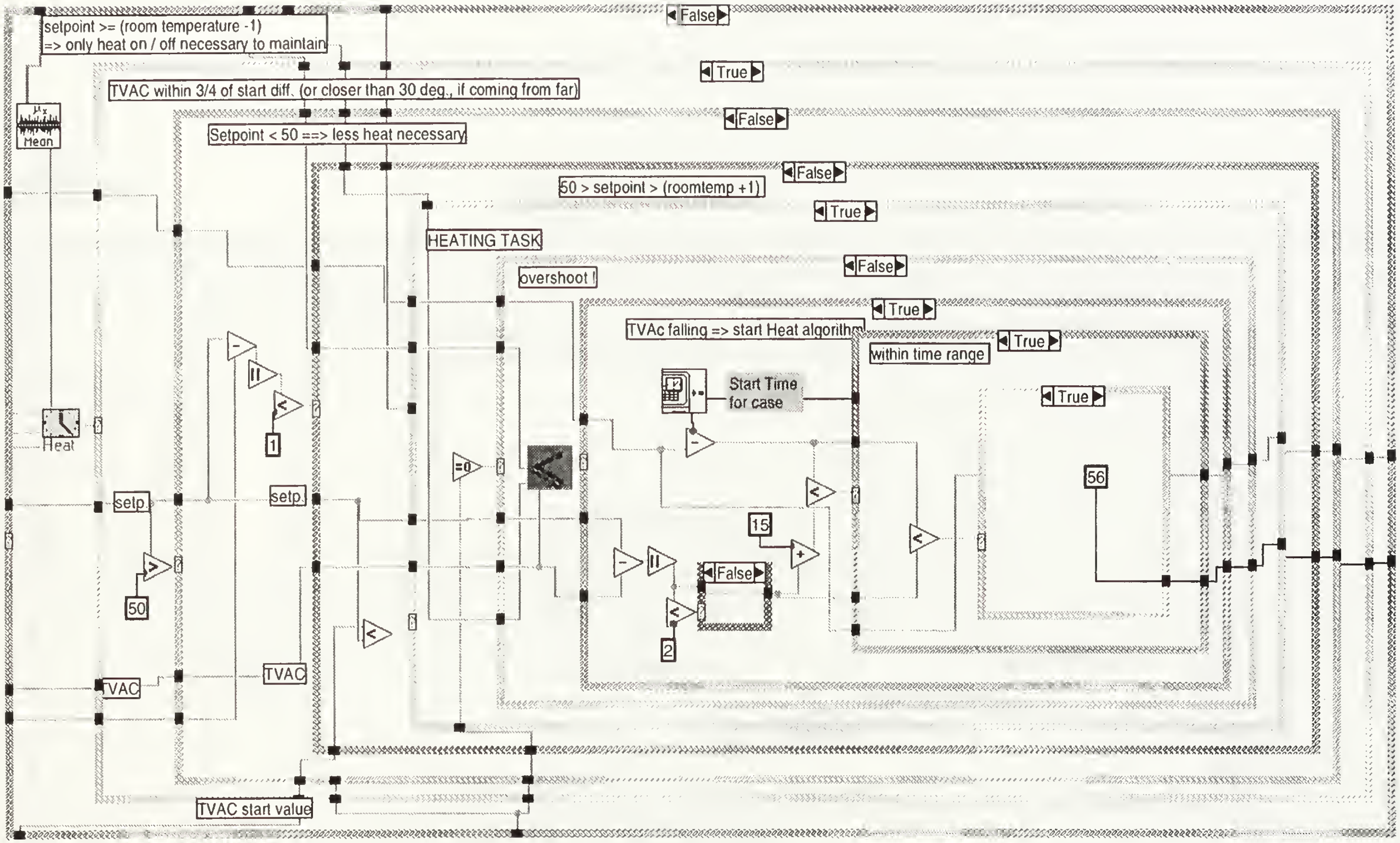

start time
0.00
overshoot
0.00
output time
0.00
setpoint
0.00
second to last TVAC
0.00
switch command
0.00
TVAC
0.00
TVAC average
0.00
CJC
0.00
array of first 15 TVAC temp.
0
0.00
temperature slope
0.00
TVAC start value
0.00

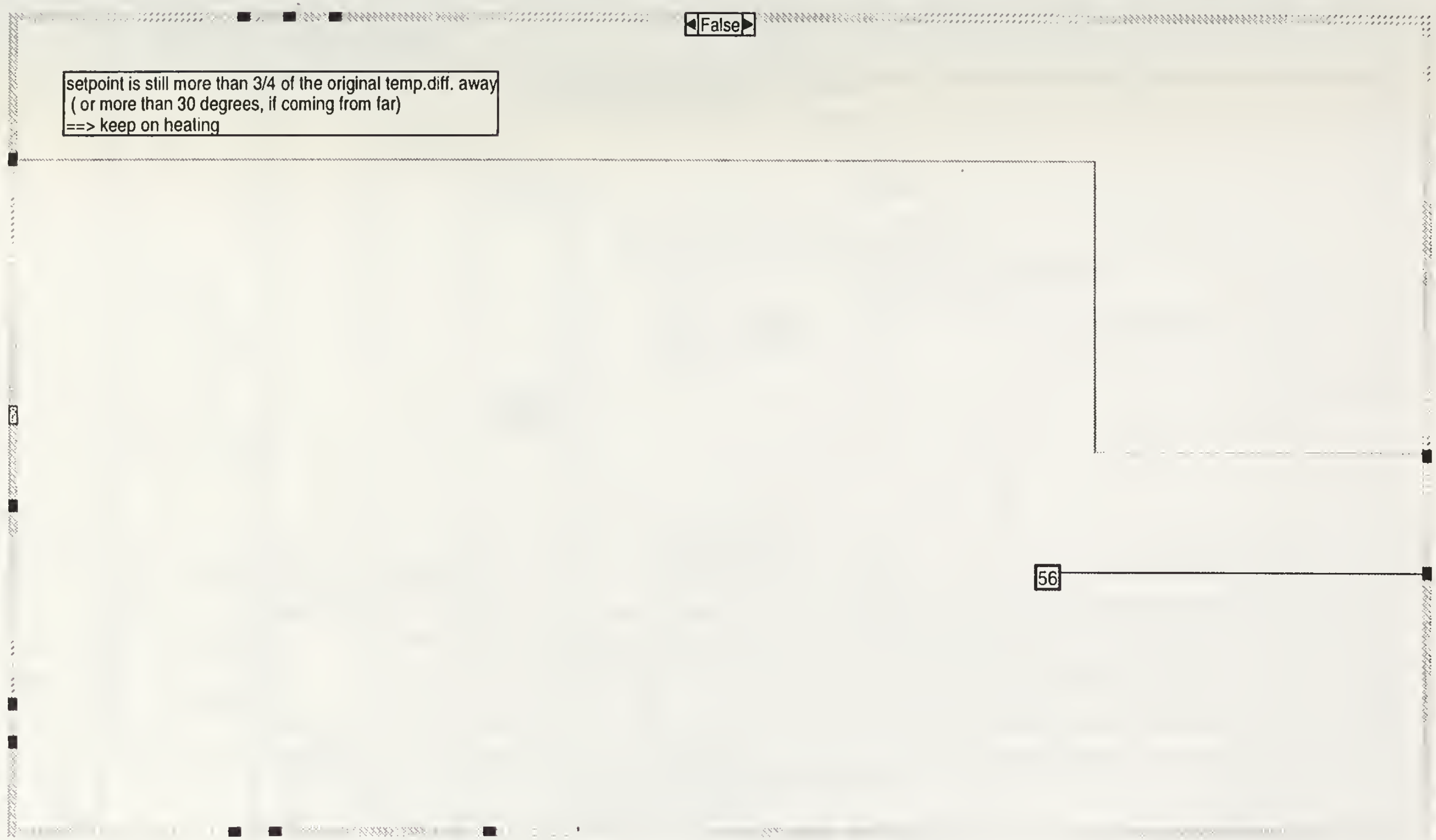
Controls and Indicators

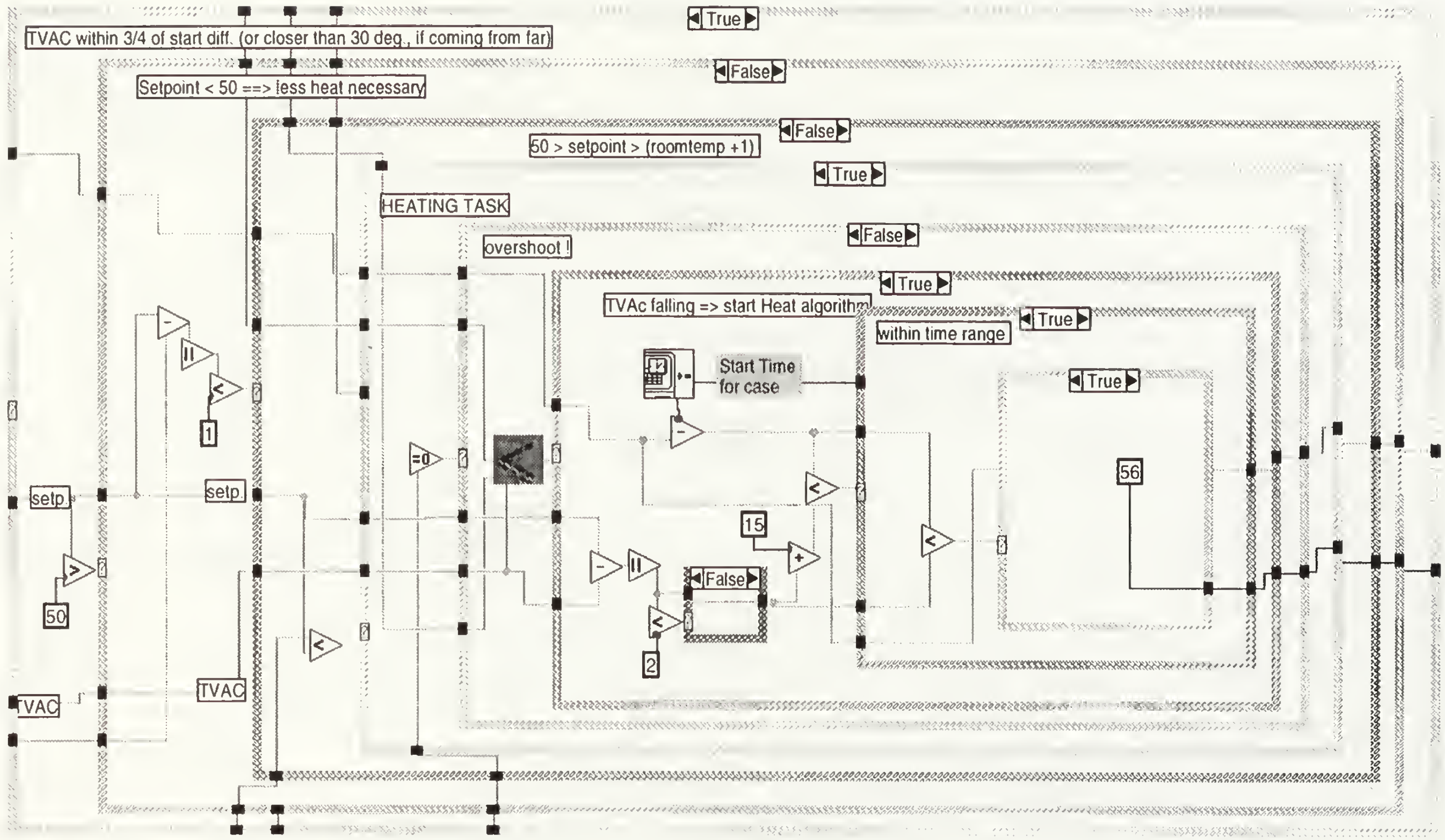
- DBL start time
- DBL setpoint
- DBL TVAC
- DBL CJC
- DBL overshoot
- DBL second to last TVAC
- DBL TVAC average
- DBL temperature slope
- [DBL] array of first 15 TVAC temp.
  - DBL
- DBL TVAC start value
- DBL output time
- DBL switch command

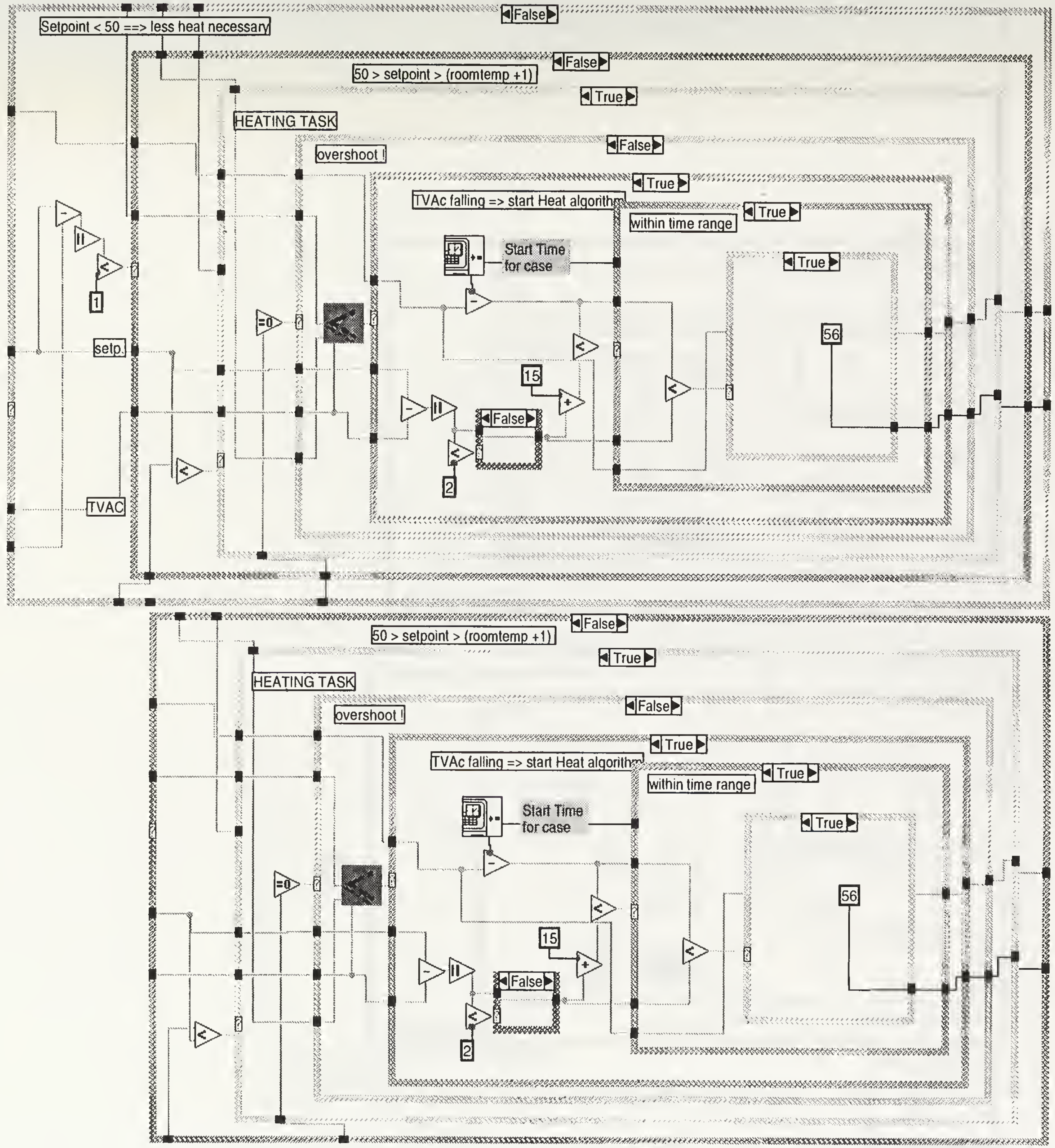
Block Diagram

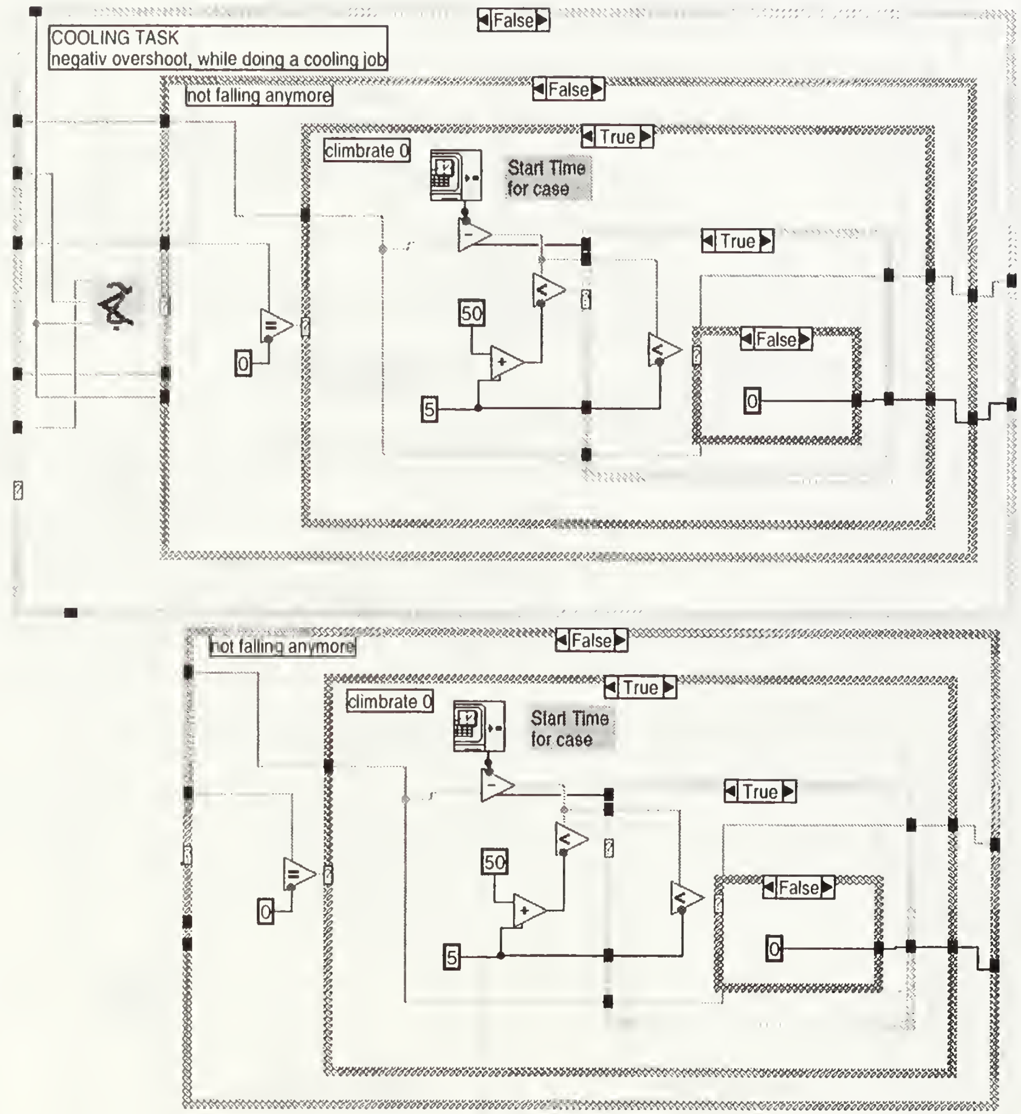


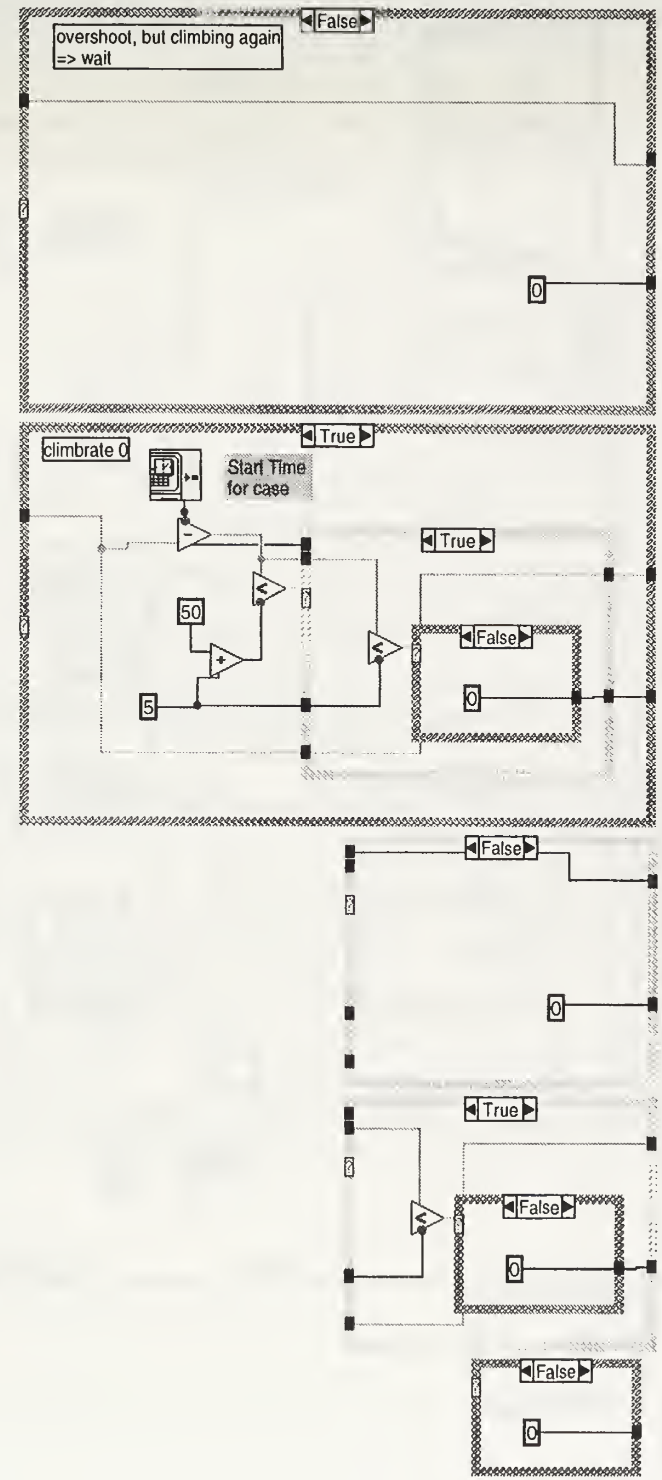

setpoint >= (room temperature -1)
=> only heat on / off necessary to maintain
TVAC within 3/4 of start diff. (or closer than 30 deg., if coming from far)
Setpoint < 50 ==> less heat necessary
50 > setpoint > (roomtemp +1)
HEATING TASK
overshoot !
TVAc falling => start Heat algorithm
within time range
Start Time
for case
False
True
False
False
True
False
True
True
True
False
Mean
Heat
setp.
setp.
TVAC
TVAC
TVAC start value
50
1
15
2
56

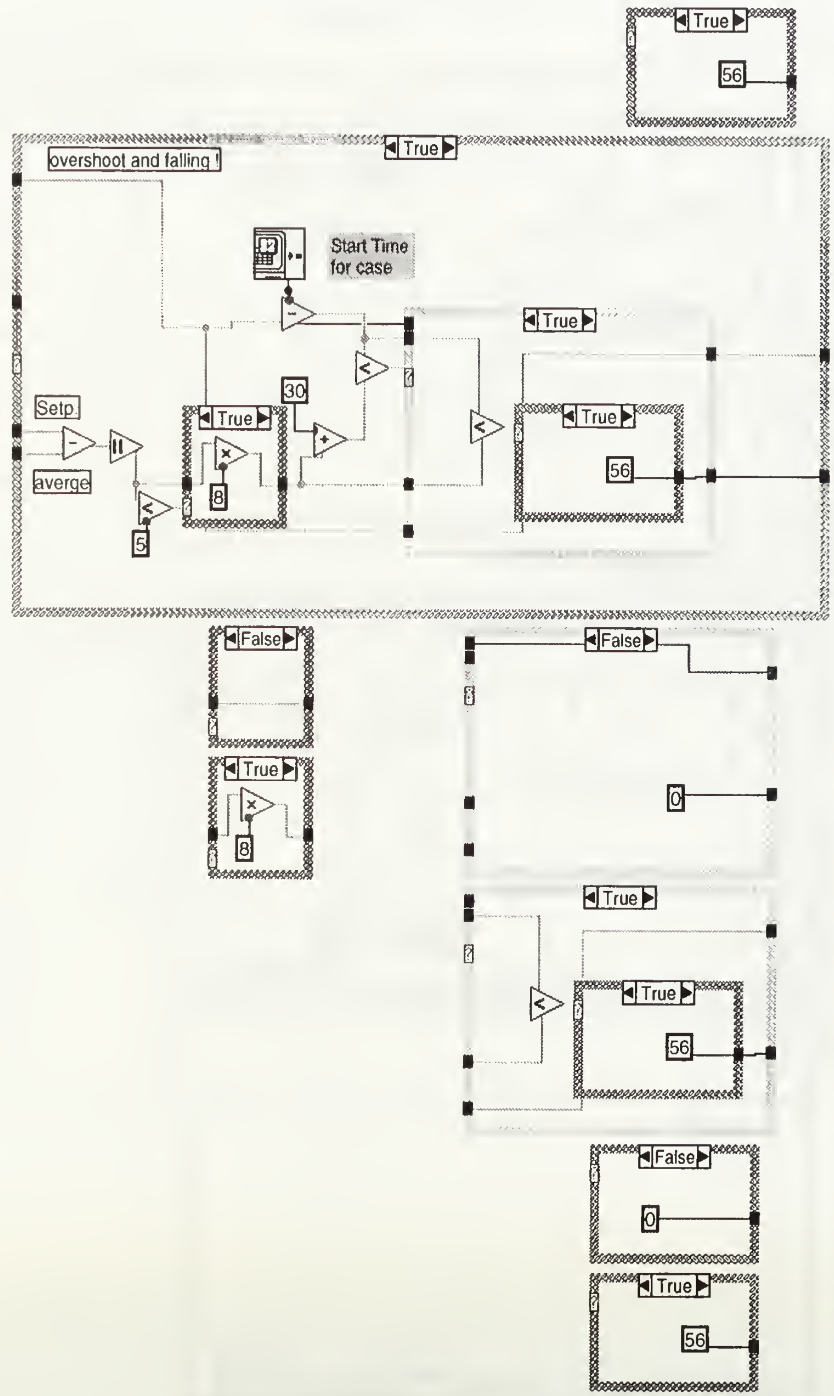

False
setpoint is still more than 3/4 of the original temp.diff. away
(or more than 30 degrees, if coming from far)
==> keep on heating
56

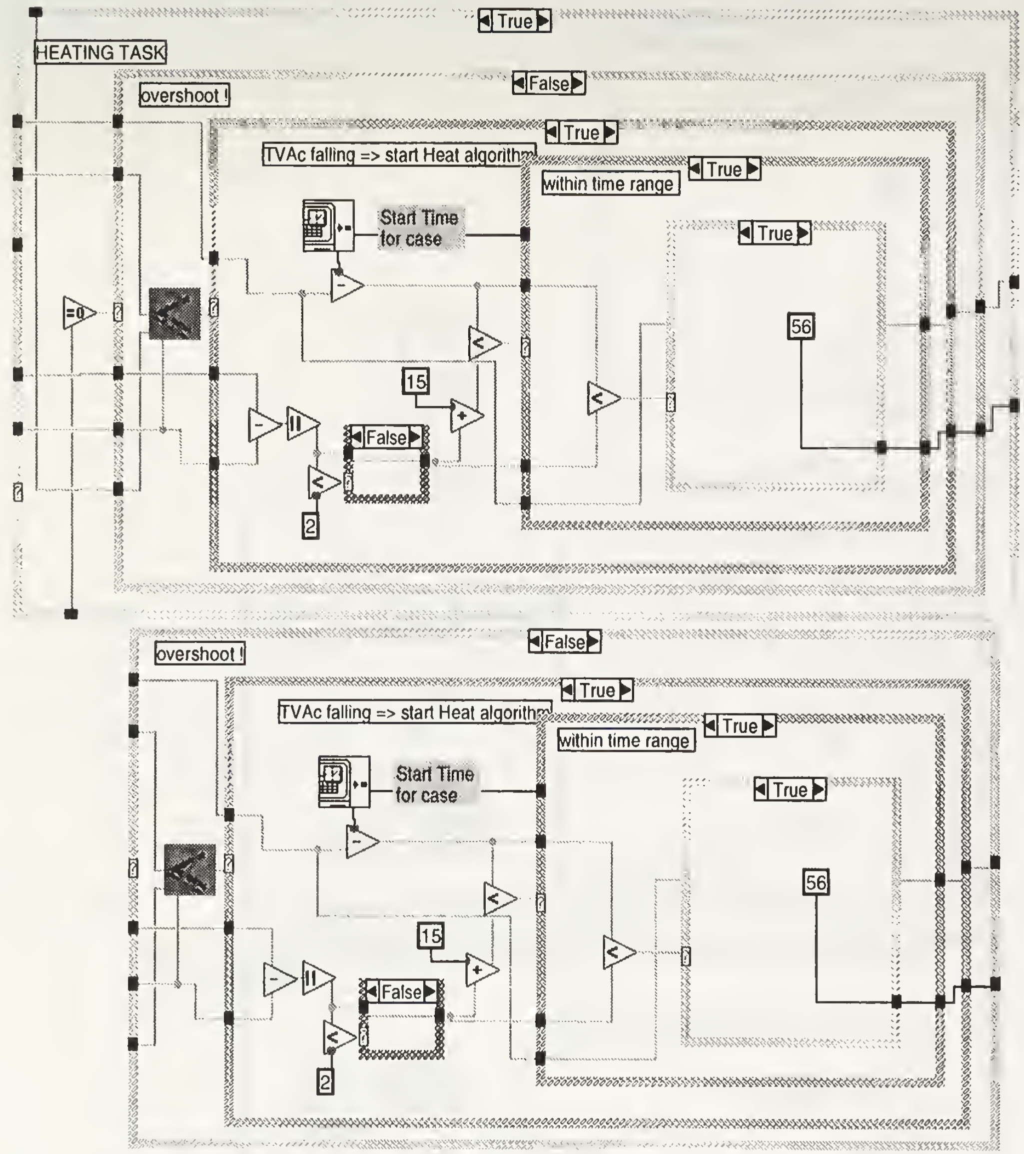

True
TVAC within 3/4 of start diff. (or closer than 30 deg., if coming from far)
False
Setpoint < 50 ==> less heat necessary
False
50 > setpoint > (roomtemp +1)
True
HEATING TASK
False
overshoot !
True
TVAc falling => start Heat algorithm
True
within time range
True
Start Time
for case
56
15
False
2
1
50
setp
setp
TVAC
TVAC

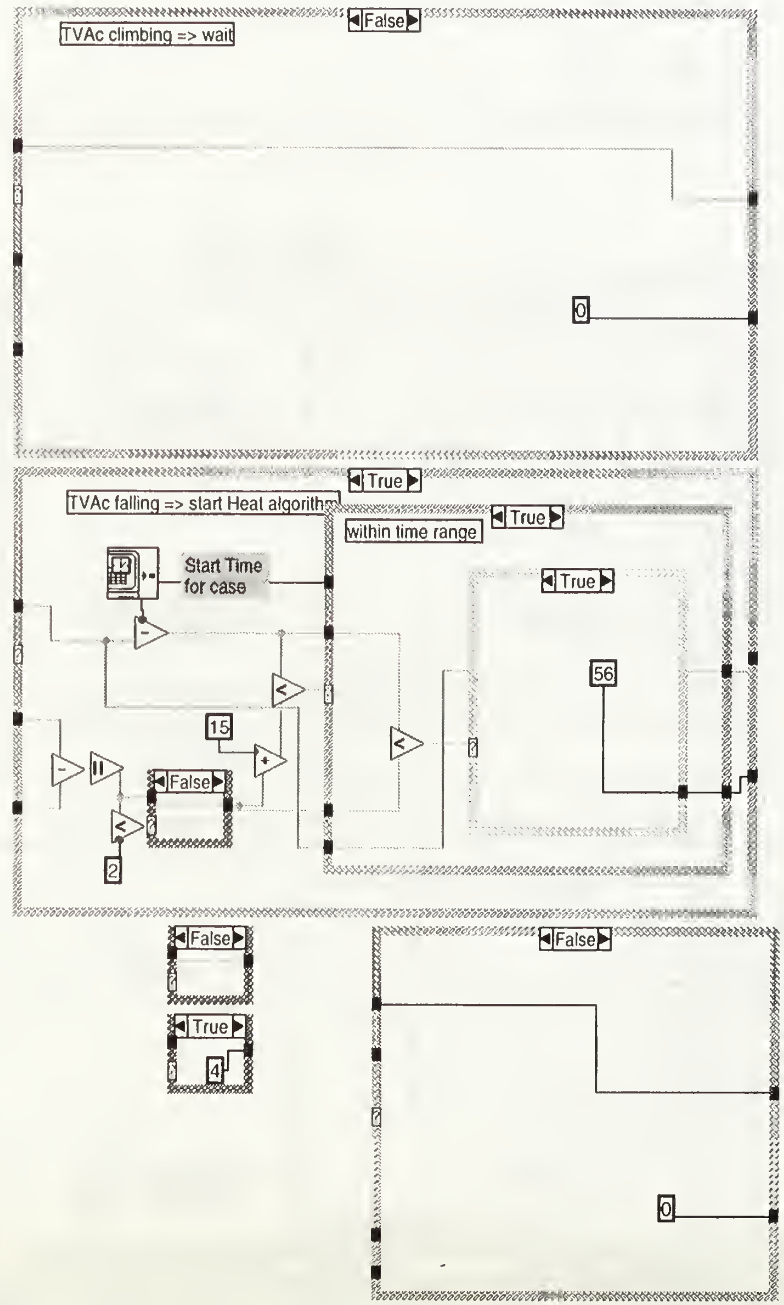

False
Setpoint < 50 ==> less heat necessary
False
50 > setpoint > (roomtemp +1)
True
HEATING TASK
False
overshoot !
True
TVAc falling => start Heat algorithm
True
within time range
True
Start Time for case
56
15
False
2
1
setp.
TVAC
False
50 > setpoint > (roomtemp +1)
True
HEATING TASK
False
overshoot !
True
TVAc falling => start Heat algorithm
True
within time range
True
Start Time for case
56
15
False
2

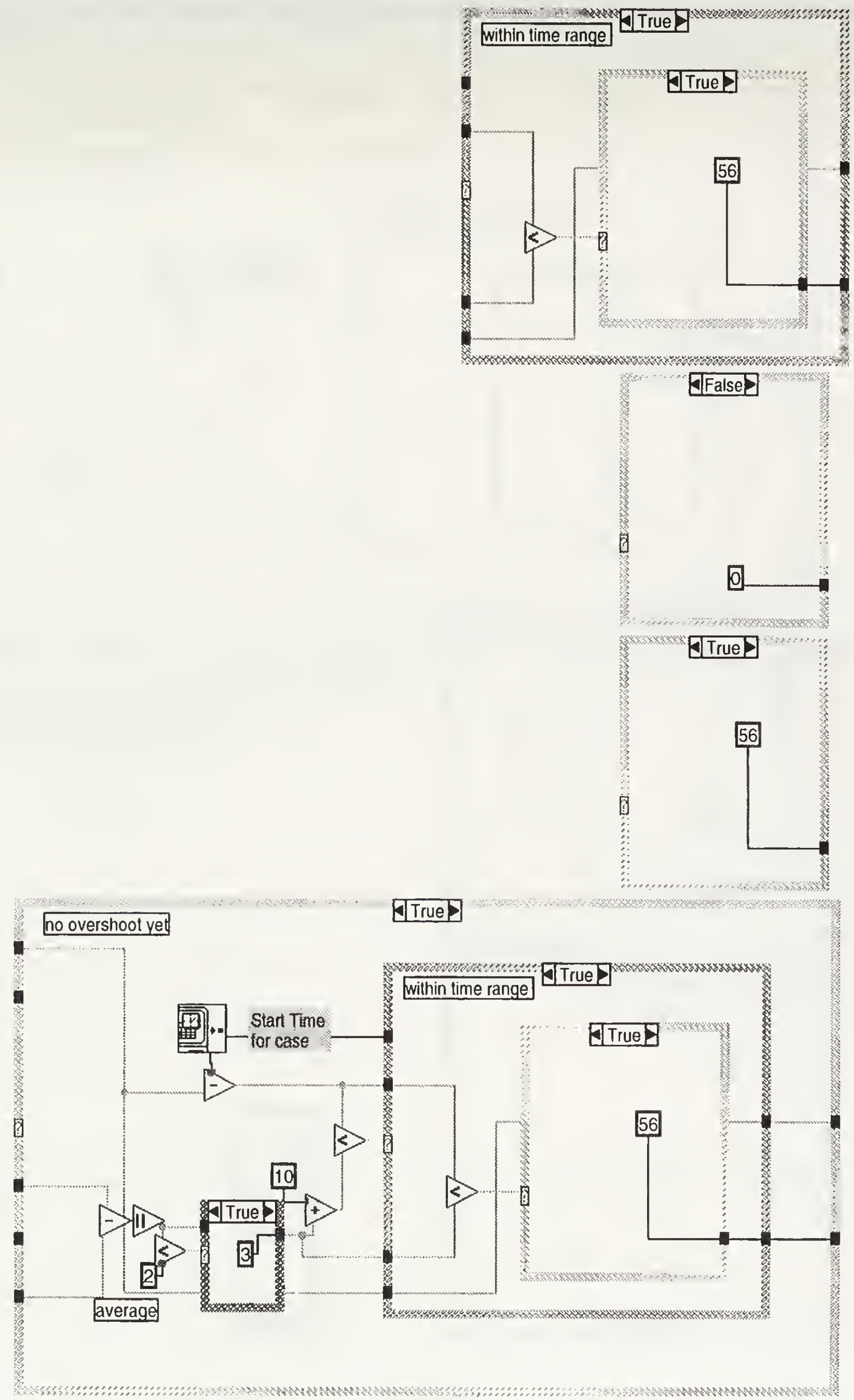

False
COOLING TASK
negativ overshoot, while doing a cooling job
False
not falling anymore
True
climbrate 0
Start Time
for case
True
50
5
0
False
0
False
not falling anymore
True
climbrate 0
Start Time
for case
True
50
5
0
False
0

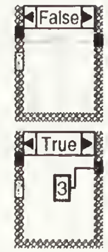

False
overshoot, but climbing again
=> wait
0
True
climbrate 0
Start Time
for case
True
50
5
False
0
False
0
True
False
0
False
0

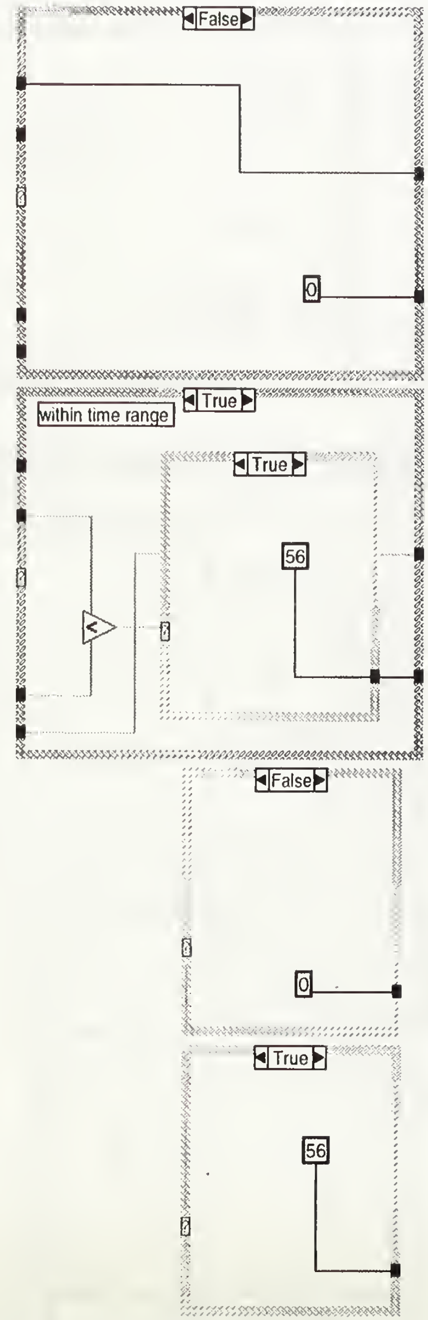

True
56
True
overshoot and falling !
Start Time
for case
Setp.
averge
5
True
8
30
True
True
56
False
True
8
False
0
True
56
False
0
True
56

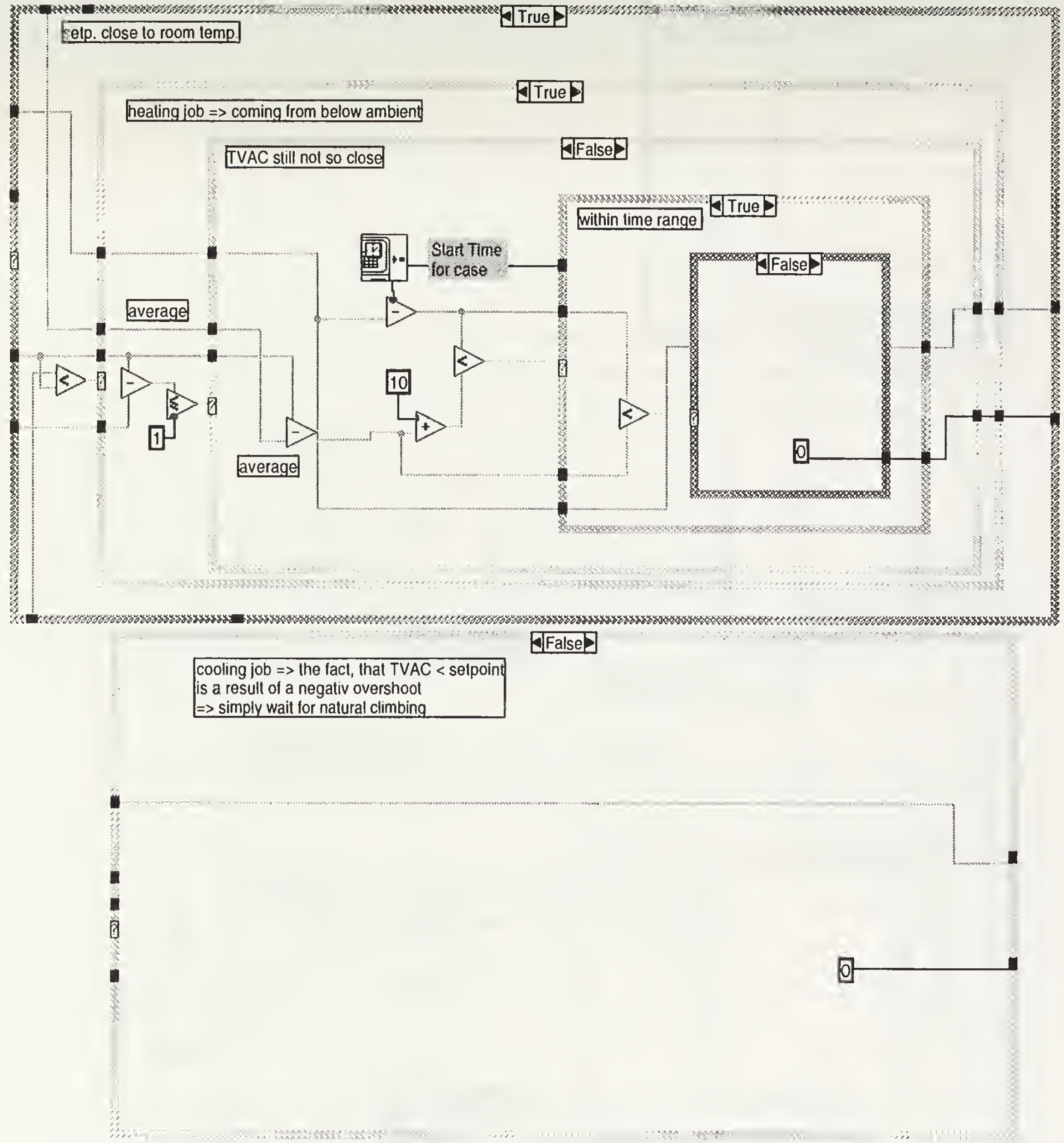

True
HEATING TASK
False
overshoot !
True
TVAc falling => start Heat algorithm
True
within time range
True
Start Time
for case
56
15
False
2
False
overshoot !
True
TVAc falling => start Heat algorithm
True
within time range
True
Start Time
for case
56
15
False
2

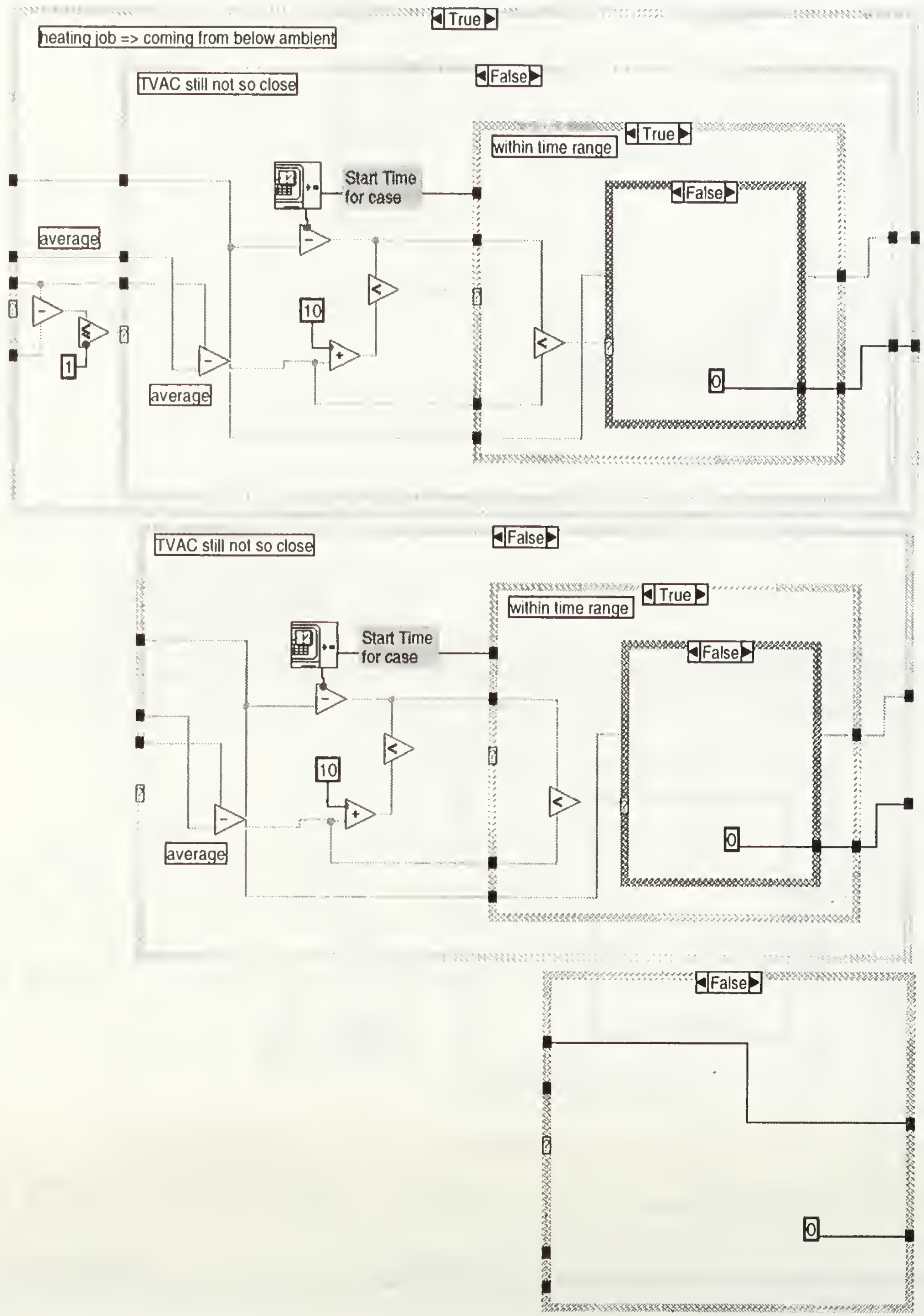

False
TVAc climbing => wait
0
True
TVAc falling => start Heat algorithm
Start Time
for case
True
within time range
True
56
15
False
2
False
True
4
False
0

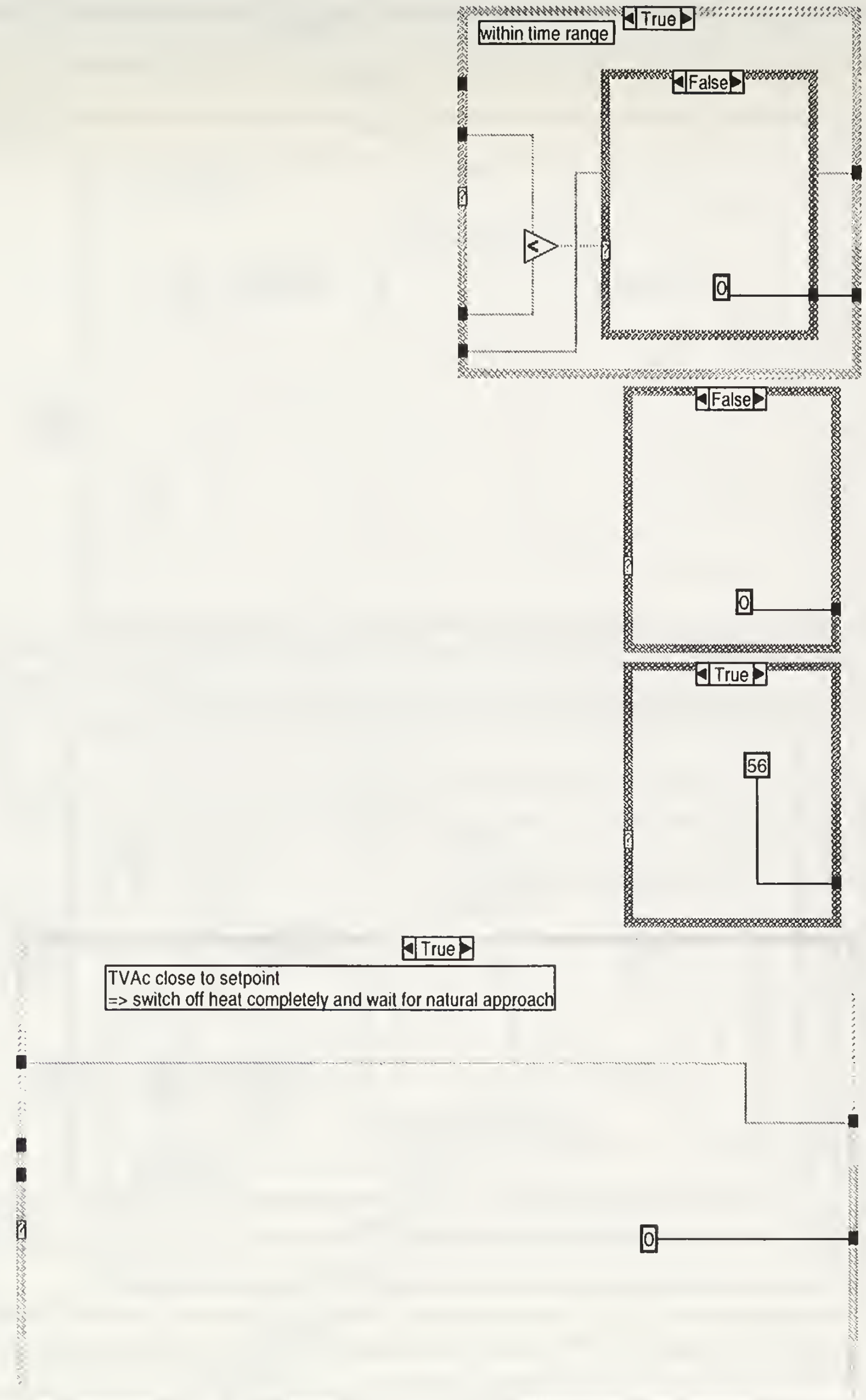

True
within time range
True
56
False
0
True
56
True
no overshoot yet
Start Time
for case
True
within time range
True
56
10
True
3
2
average

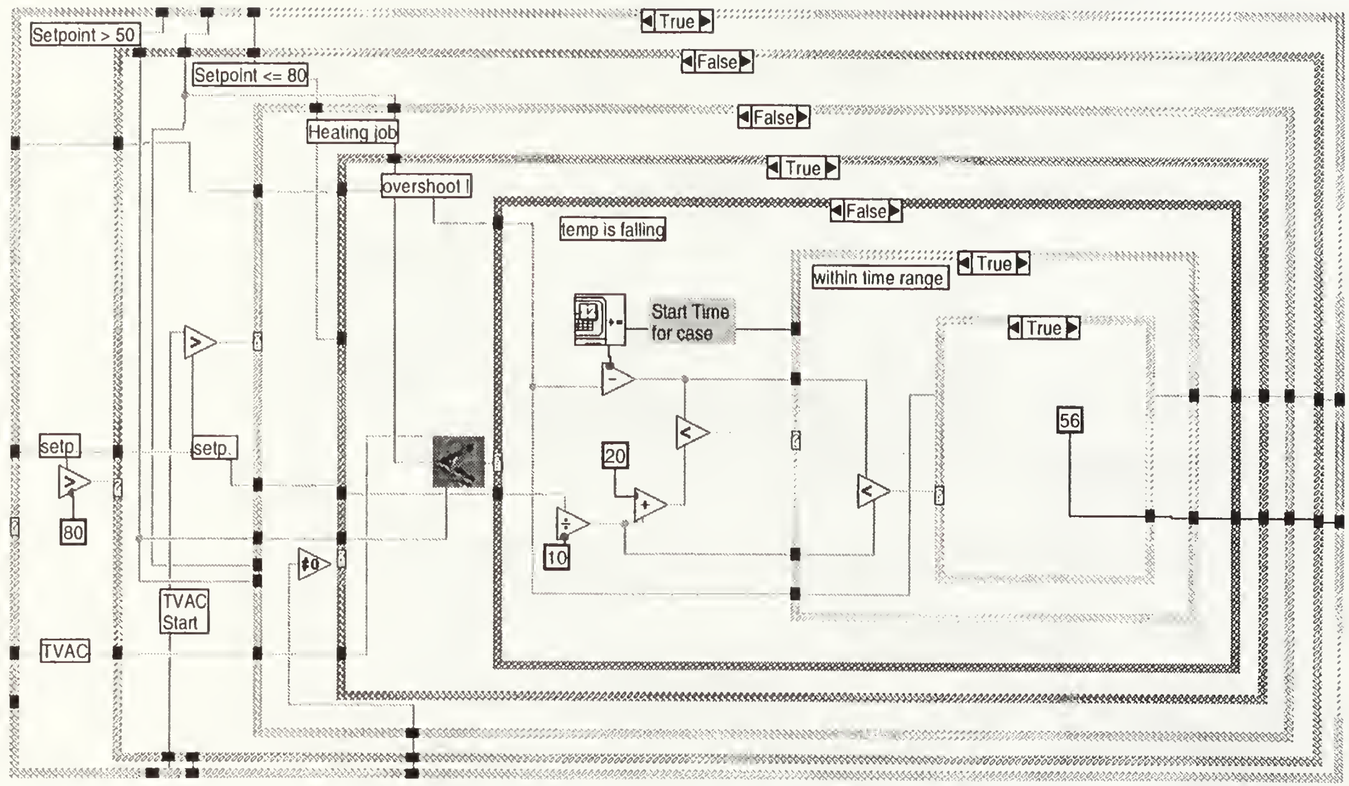

False
True
3

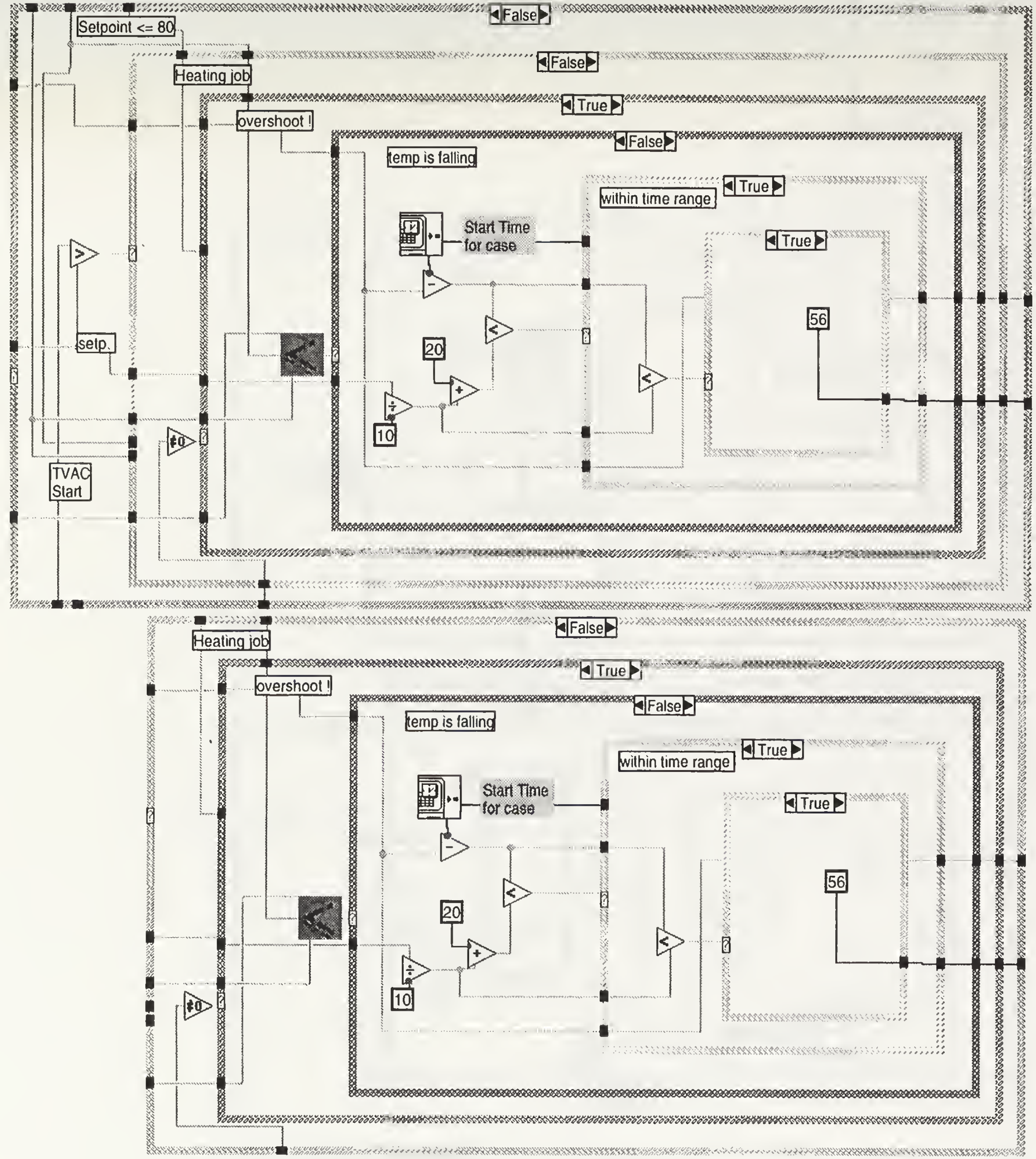

False
0
True
within time range
True
56
False
0
True
56

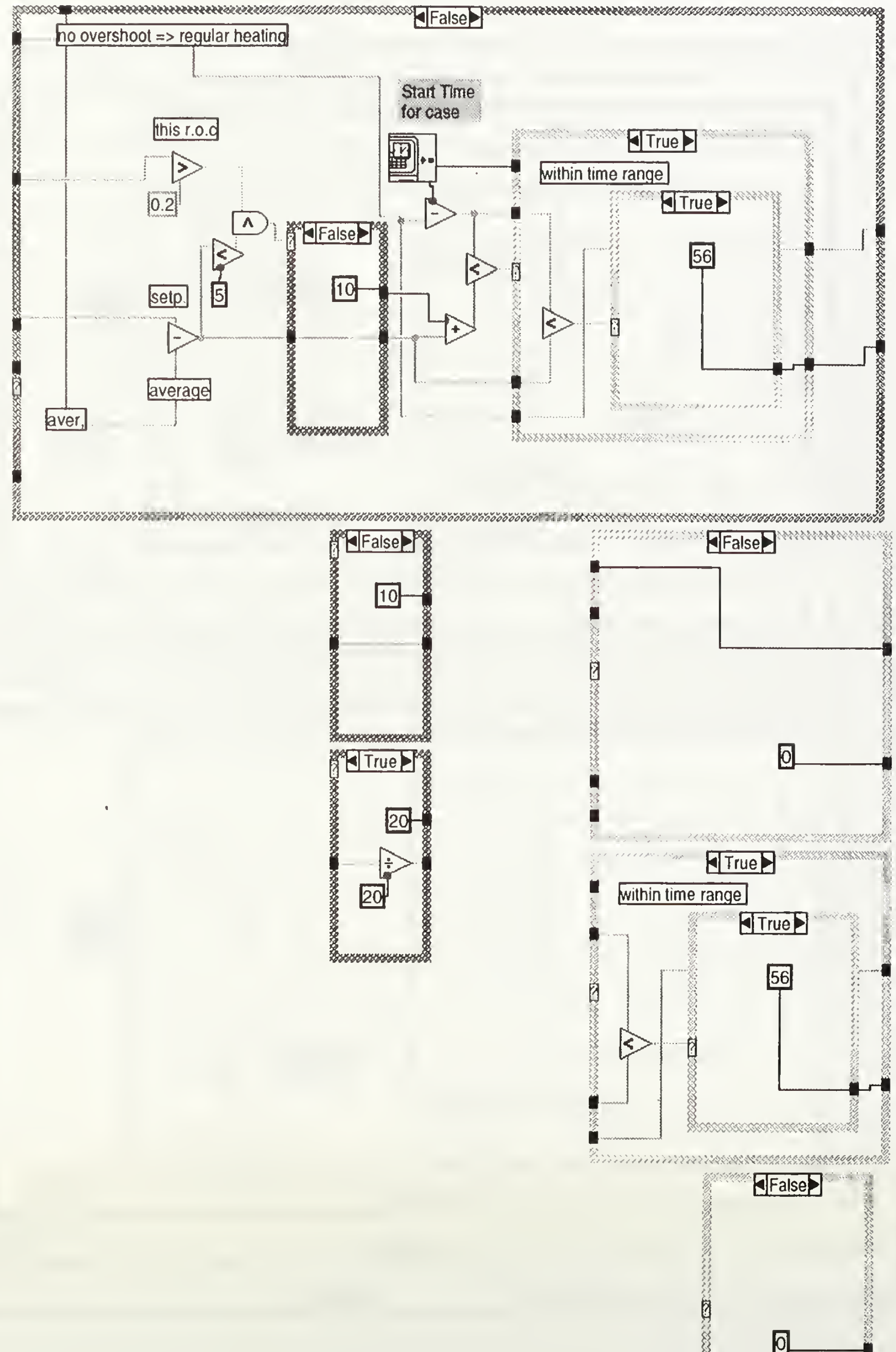

True
setp. close to room temp.
True
heating job => coming from below ambient
False
TVAC still not so close
True
within time range
Start Time
for case
average
average
10
1
False
0
False
cooling job => the fact, that TVAC < setpoint
is a result of a negativ overshoot
=> simply wait for natural climbing
0

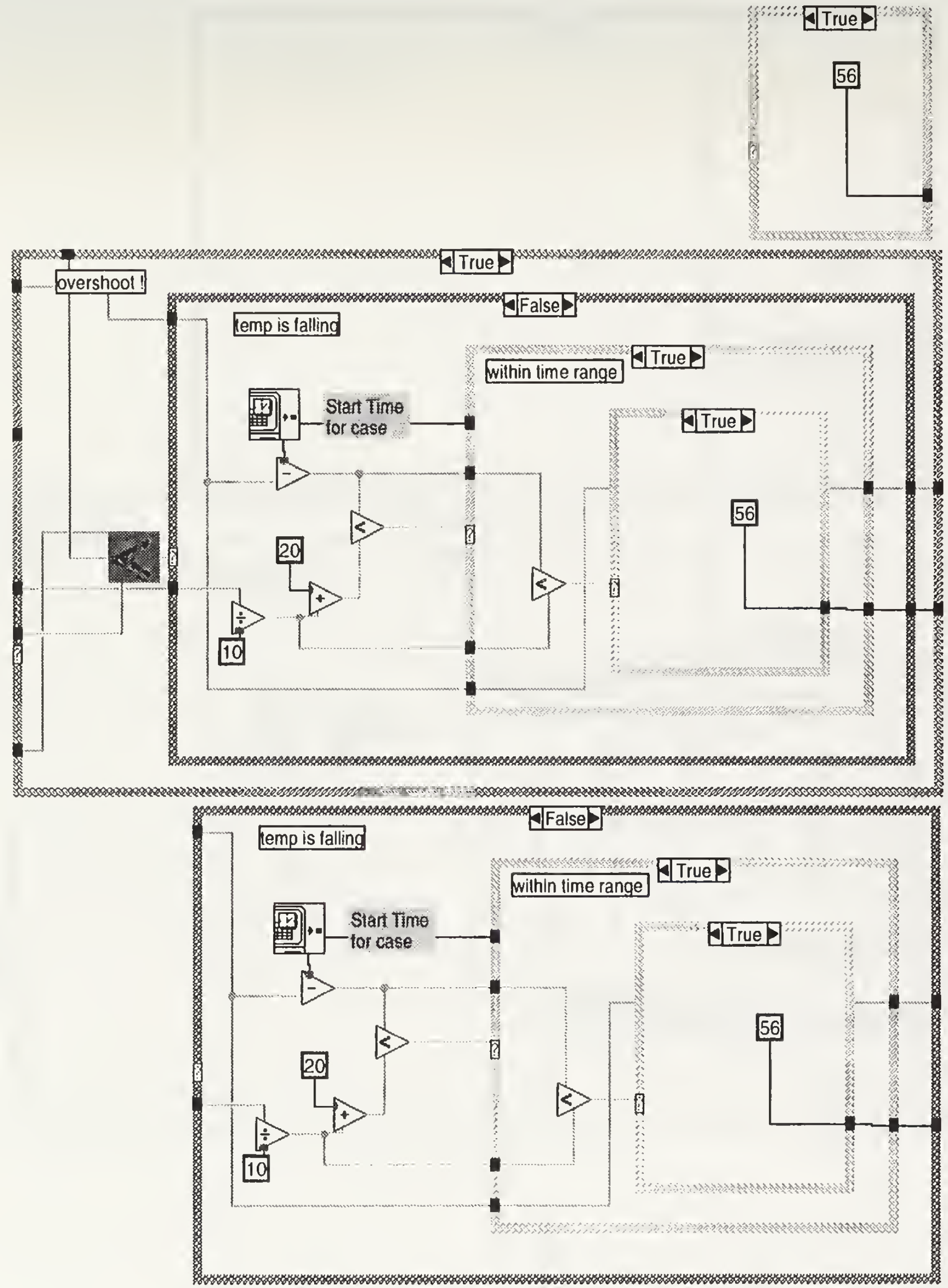

True
heating job => coming from below ambient
False
TVAC still not so close
True
within time range
False
Start Time
for case
average
10
1
average
0
False
TVAC still not so close
True
within time range
False
Start Time
for case
10
average
0
False
0

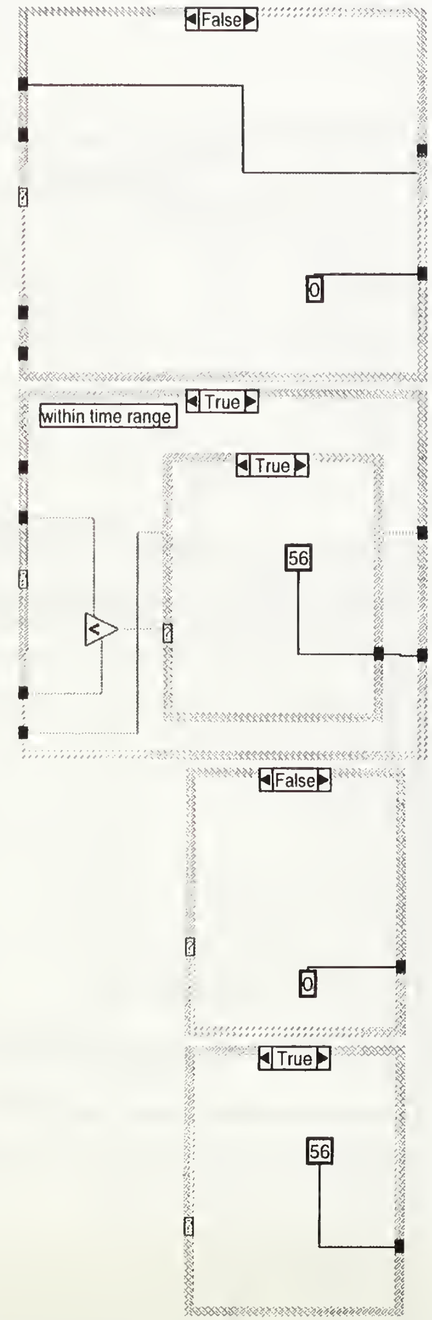

True
within time range
False
0
False
0
True
56
True
TVAc close to setpoint
=> switch off heat completely and wait for natural approach
0

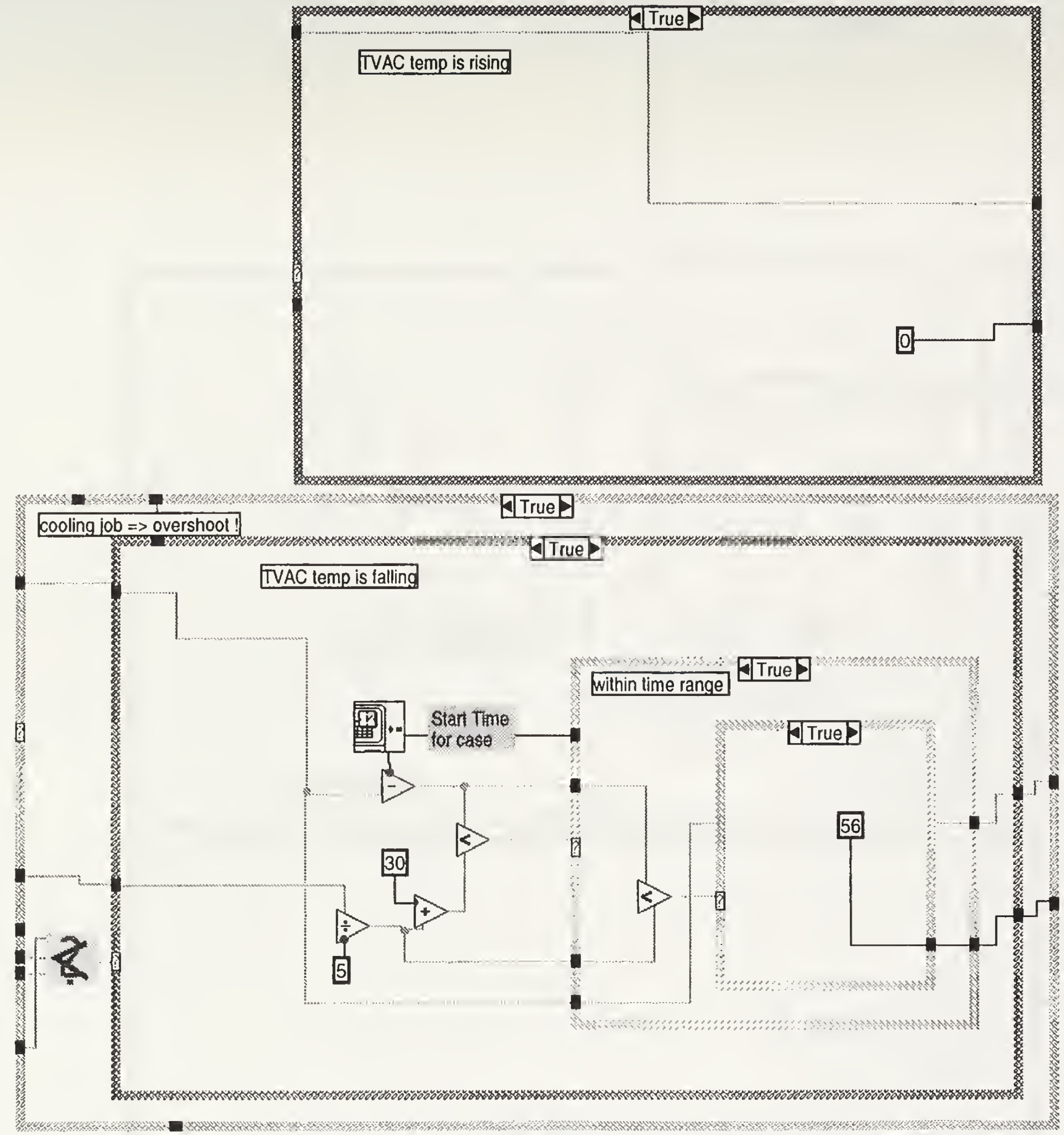

True
Setpoint > 50
False
Setpoint <= 80
False
Heating job
True
overshoot !
False
temp is falling
within time range
True
Start Time
for case
True
56
20
10
setp.
setp.
80
TVAC
Start
TVAC

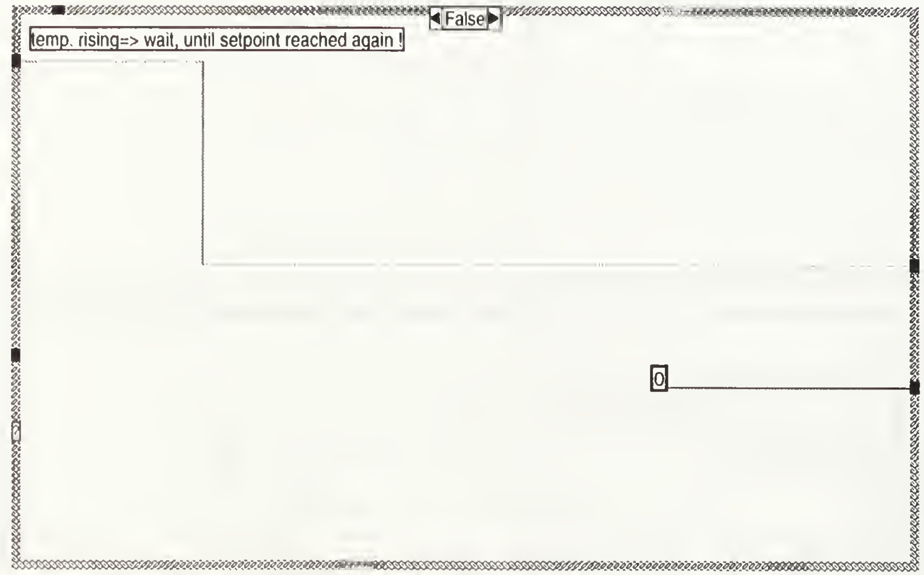

False
Setpoint <= 80
False
Heating job
True
overshoot !
False
temp is falling
within time range
True
Start Time
for case
True
56
20
10
setp.
TVAC
Start
≠0
False
Heating job
True
overshoot !
False
temp is falling
within time range
True
Start Time
for case
True
56
20
10
≠0

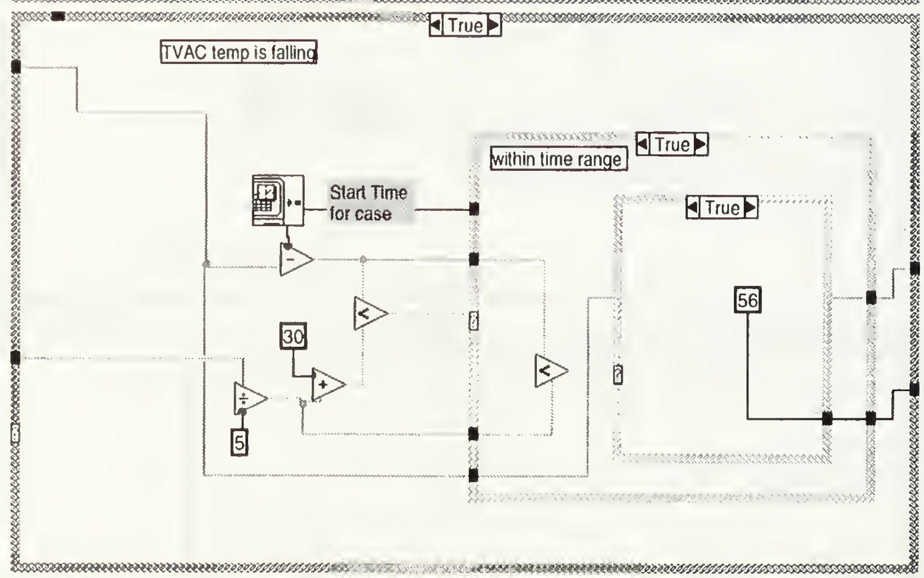

False
no overshoot => regular heating
Start Time
for case
this r.o.c
0.2
setp.
5
average
aver.
False
10
True
within time range
True
56
False
10
True
20
20
False
0
True
within time range
True
56
False
0

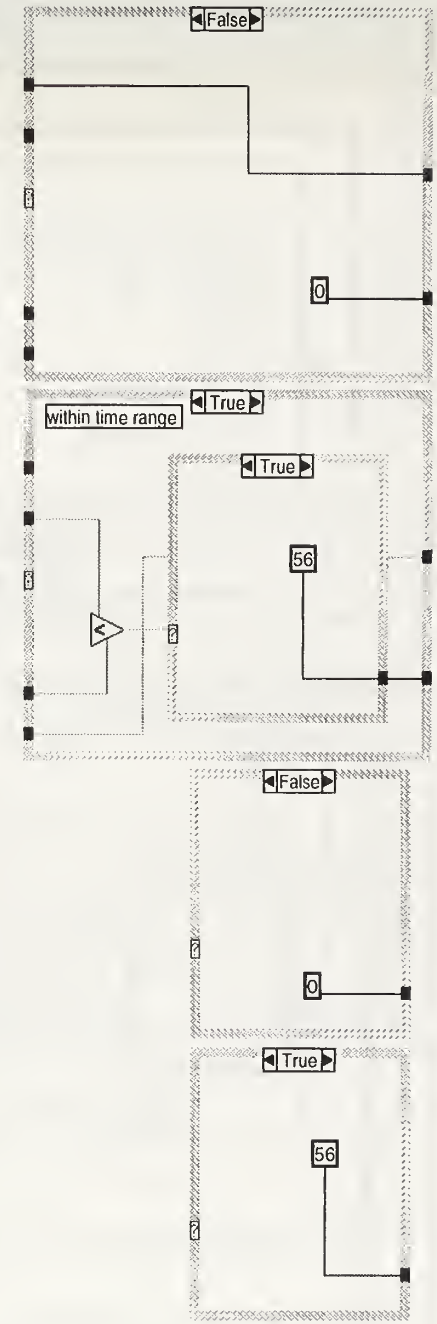

True
56
True
overshoot !
False
temp is falling
within time range
True
Start Time
for case
True
56
20
10
False
temp is falling
within time range
True
Start Time
for case
True
56
20
10

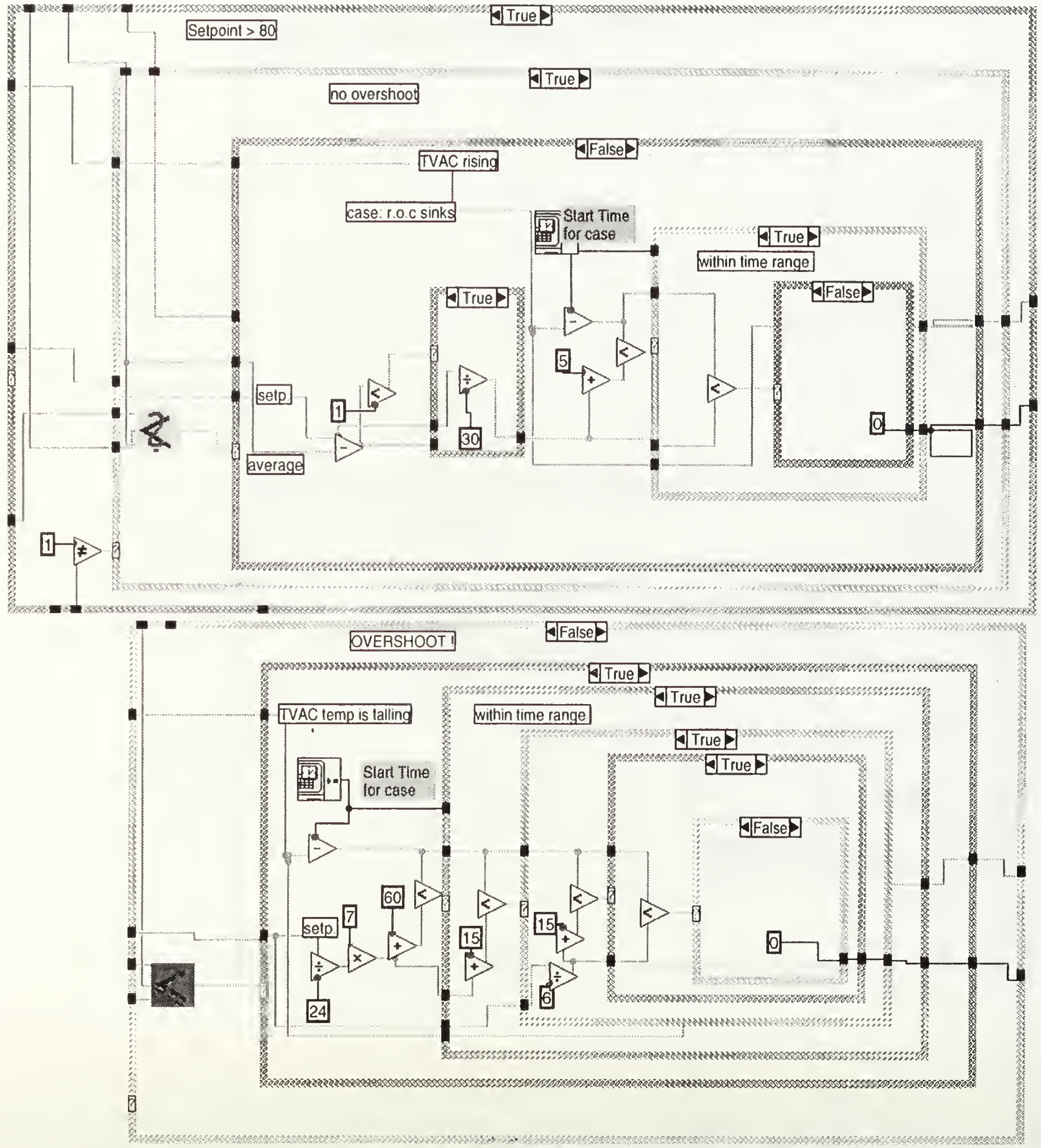

False
0
True
within time range
True
56
False
0
True
56

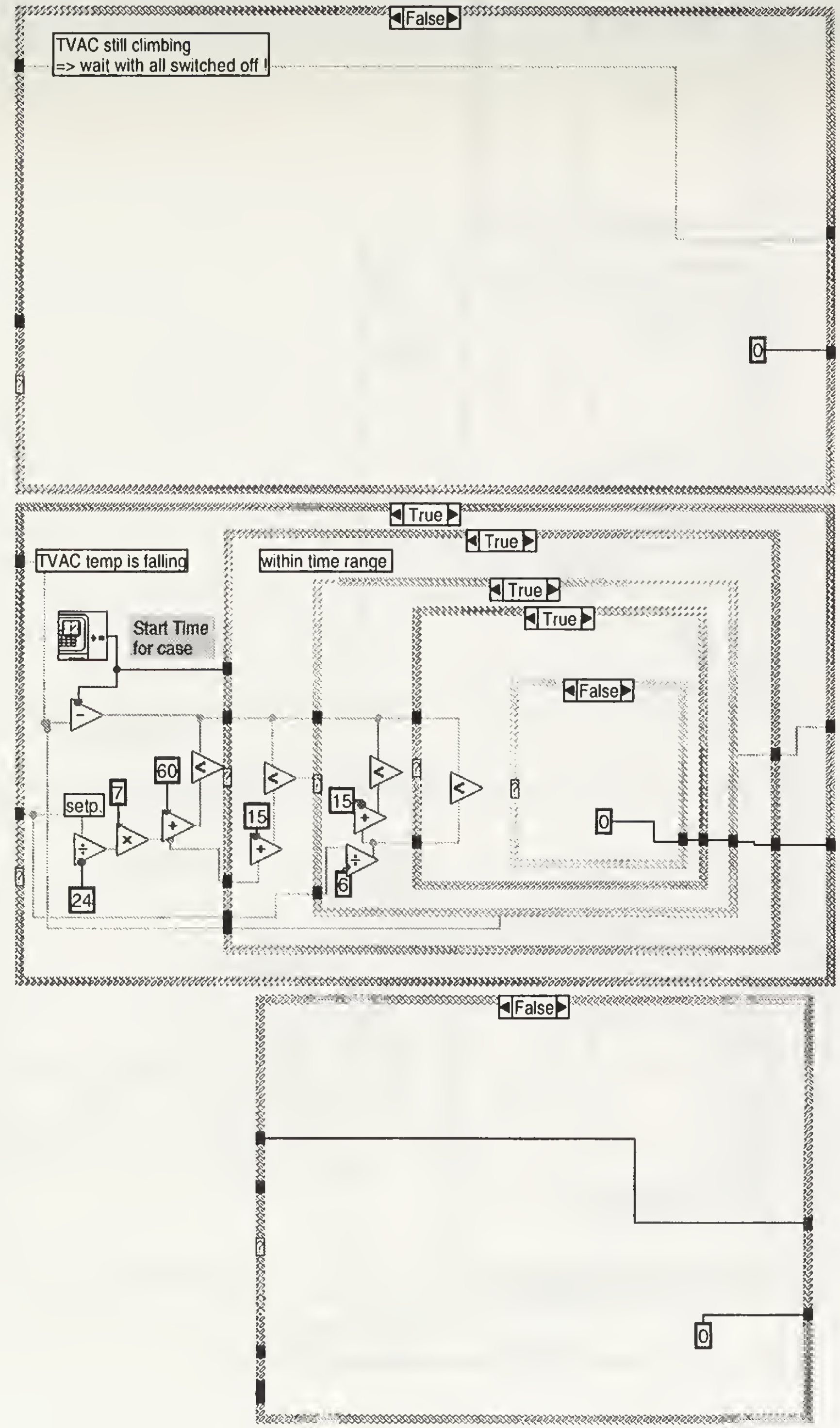

True
TVAC temp is rising
0
True
cooling job => overshoot !
True
TVAC temp is falling
Start Time
for case
30
5
within time range
True
True
56

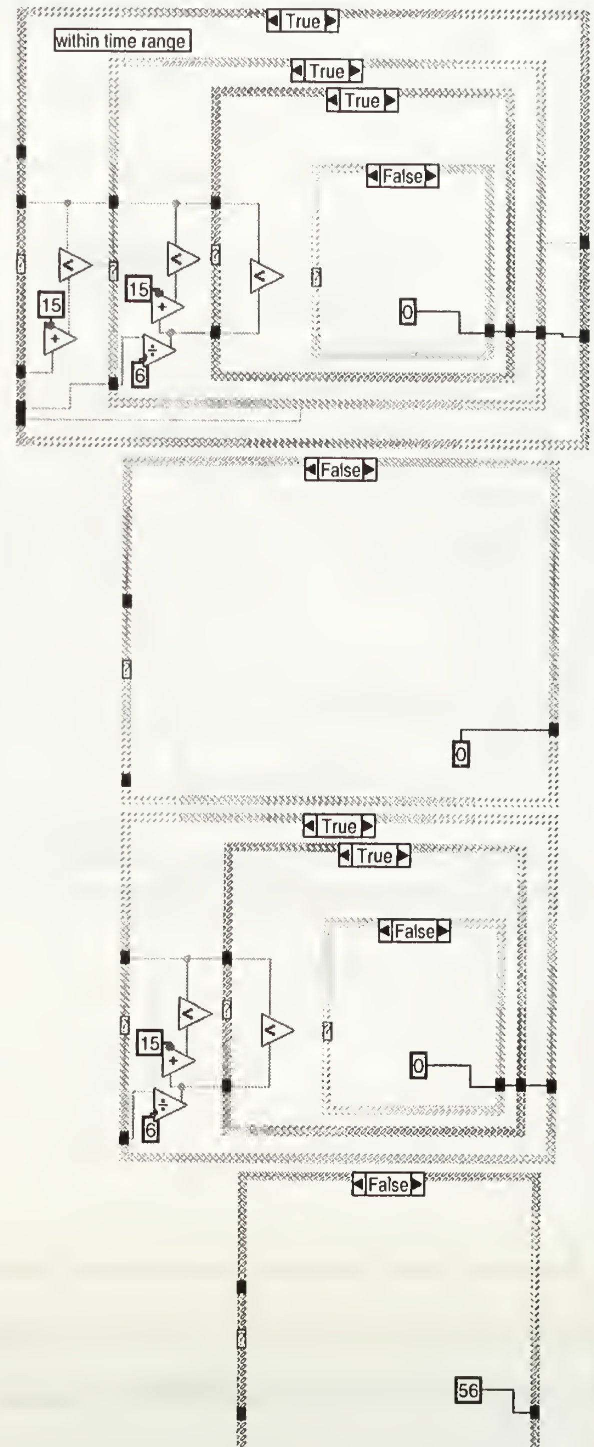

False
temp. rising=> wait, until setpoint reached again !
0

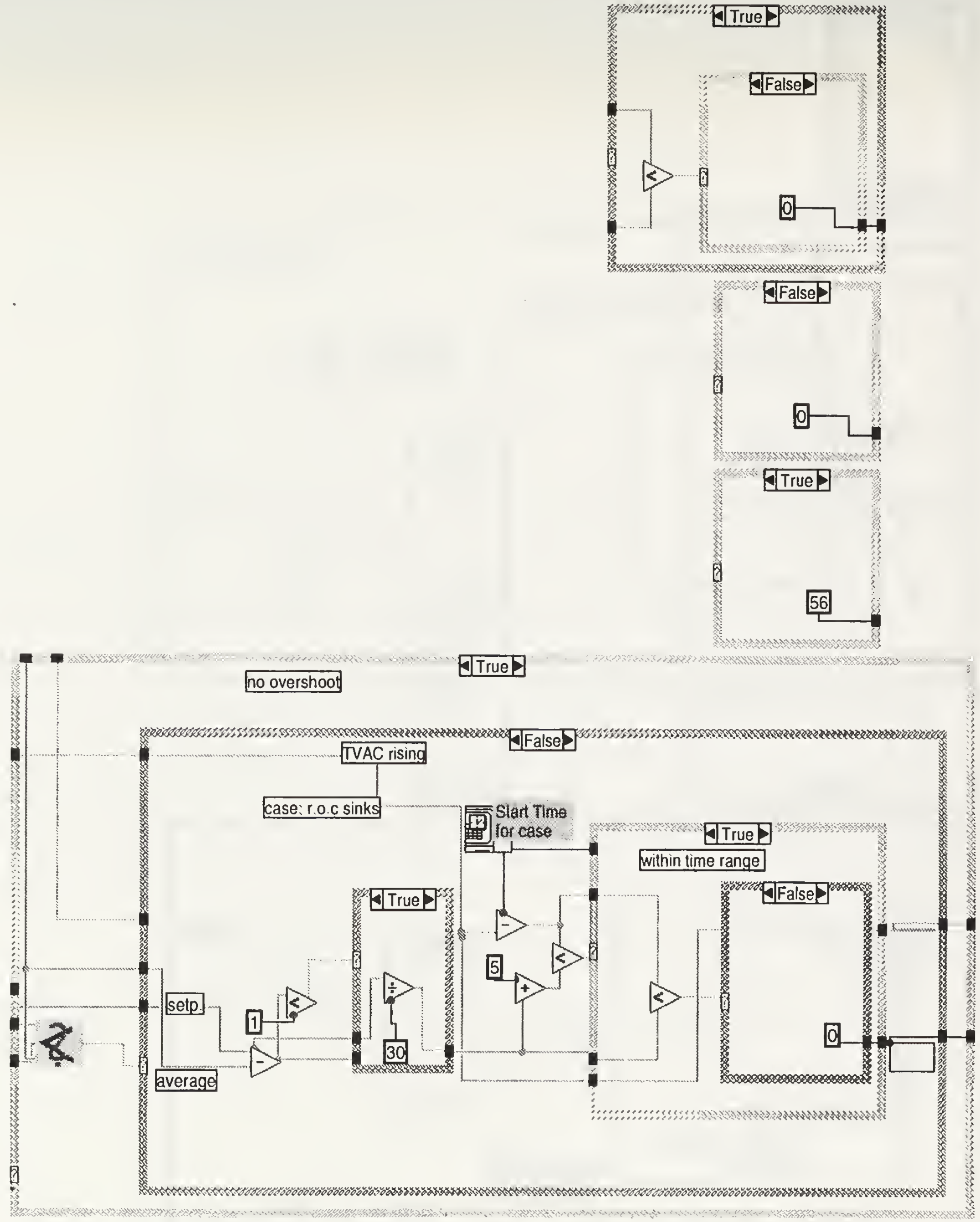

True
TVAC temp is falling
Start Time
for case
within time range
True
True
30
5
56

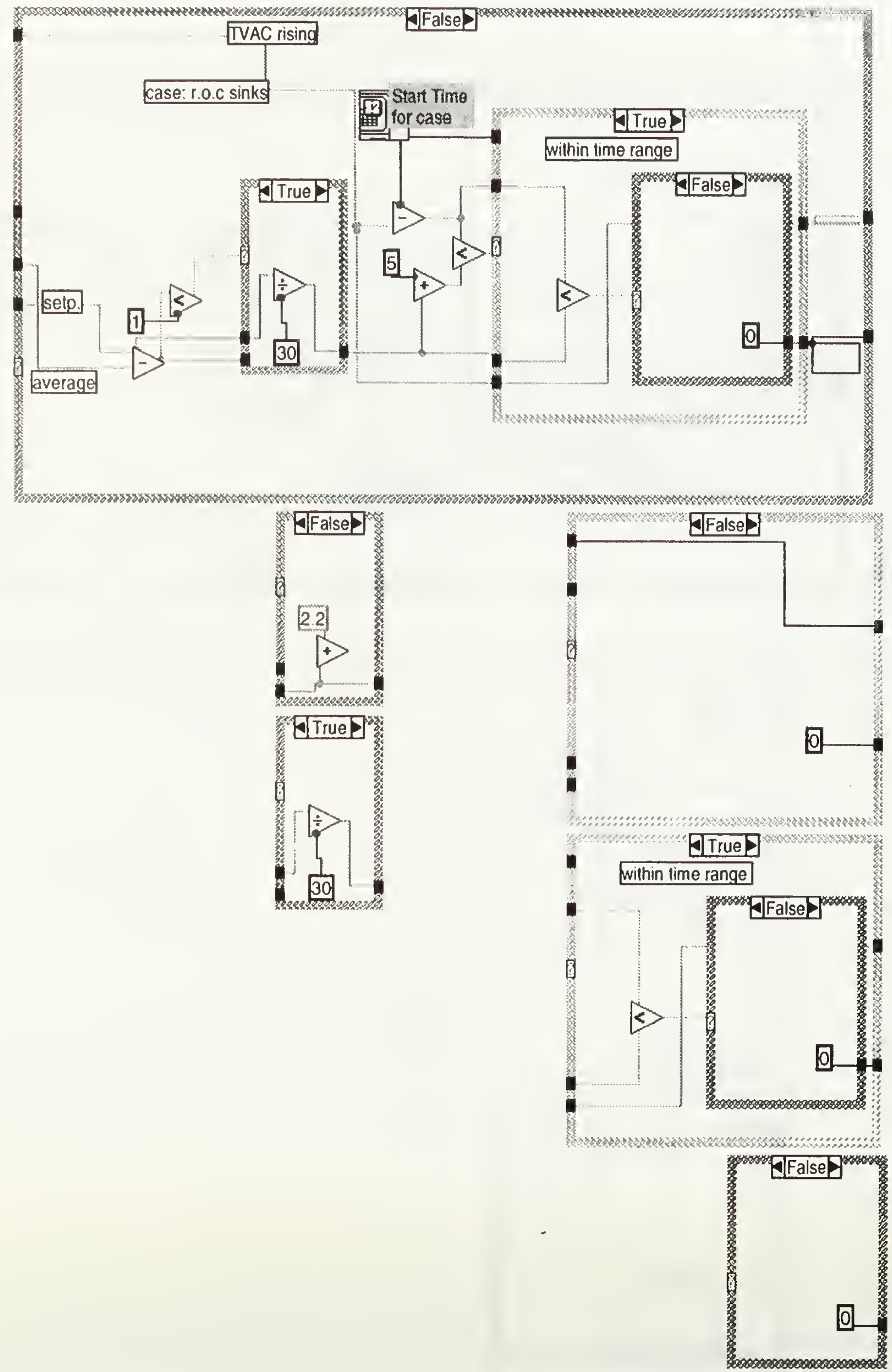

False
0
True
within time range
True
56
False
0
True
56

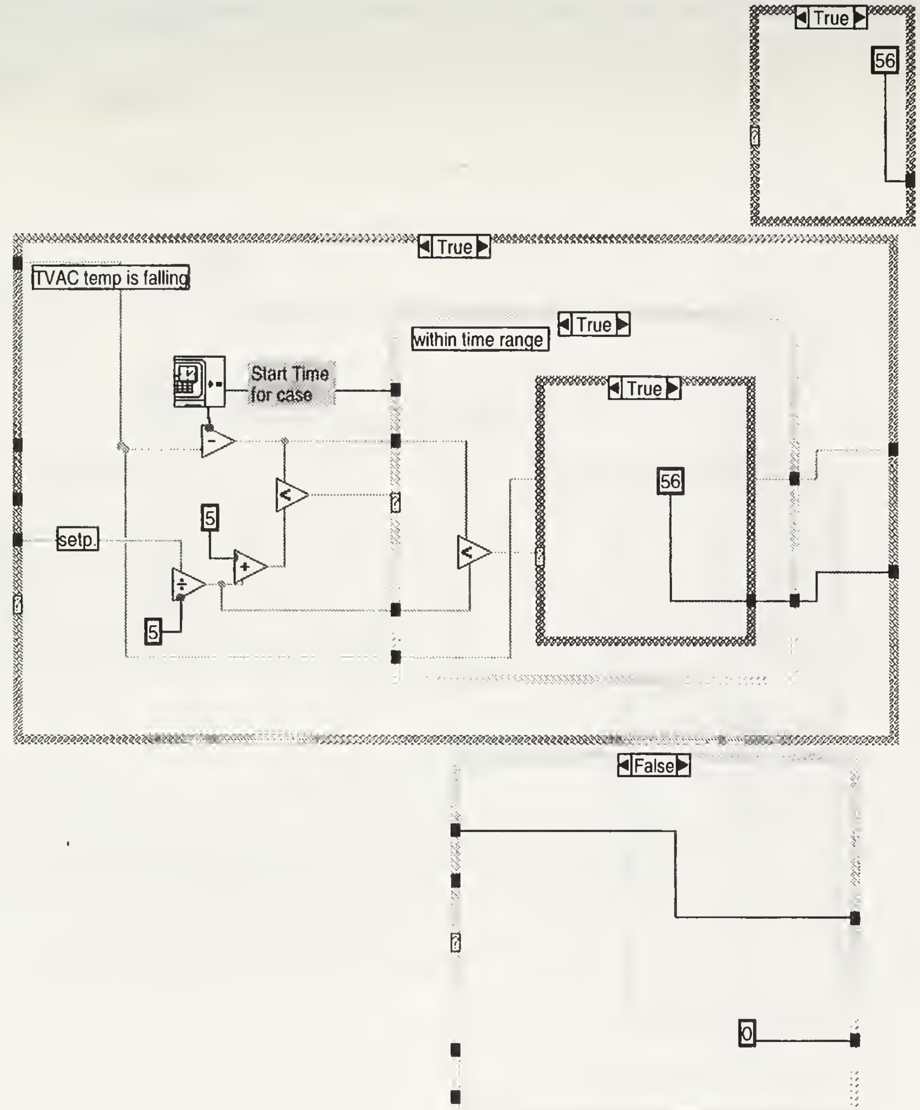

True
Setpoint > 80
True
no overshoot
False
TVAC rising
case: r.o.c sinks
Start Time
for case
True
within time range
False
True
setp.
1
average
30
5
0
1
False
OVERSHOOT !
True
True
TVAC temp is falling
within time range
True
True
Start Time
for case
False
setp.
7
60
15
15
6
24
0

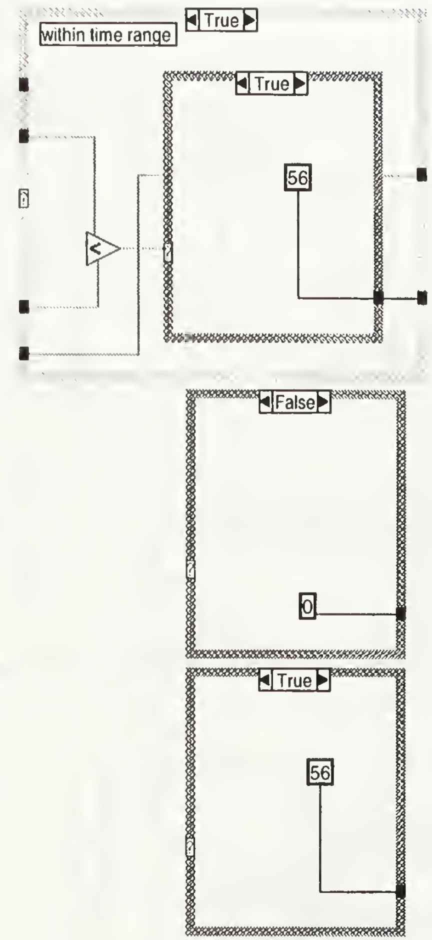

False
TVAC still climbing
=> wait with all switched off !
0
True
TVAC temp is falling
Start Time
for case
setp.
7
24
60
True
within time range
15
True
15
6
True
False
0
False
0

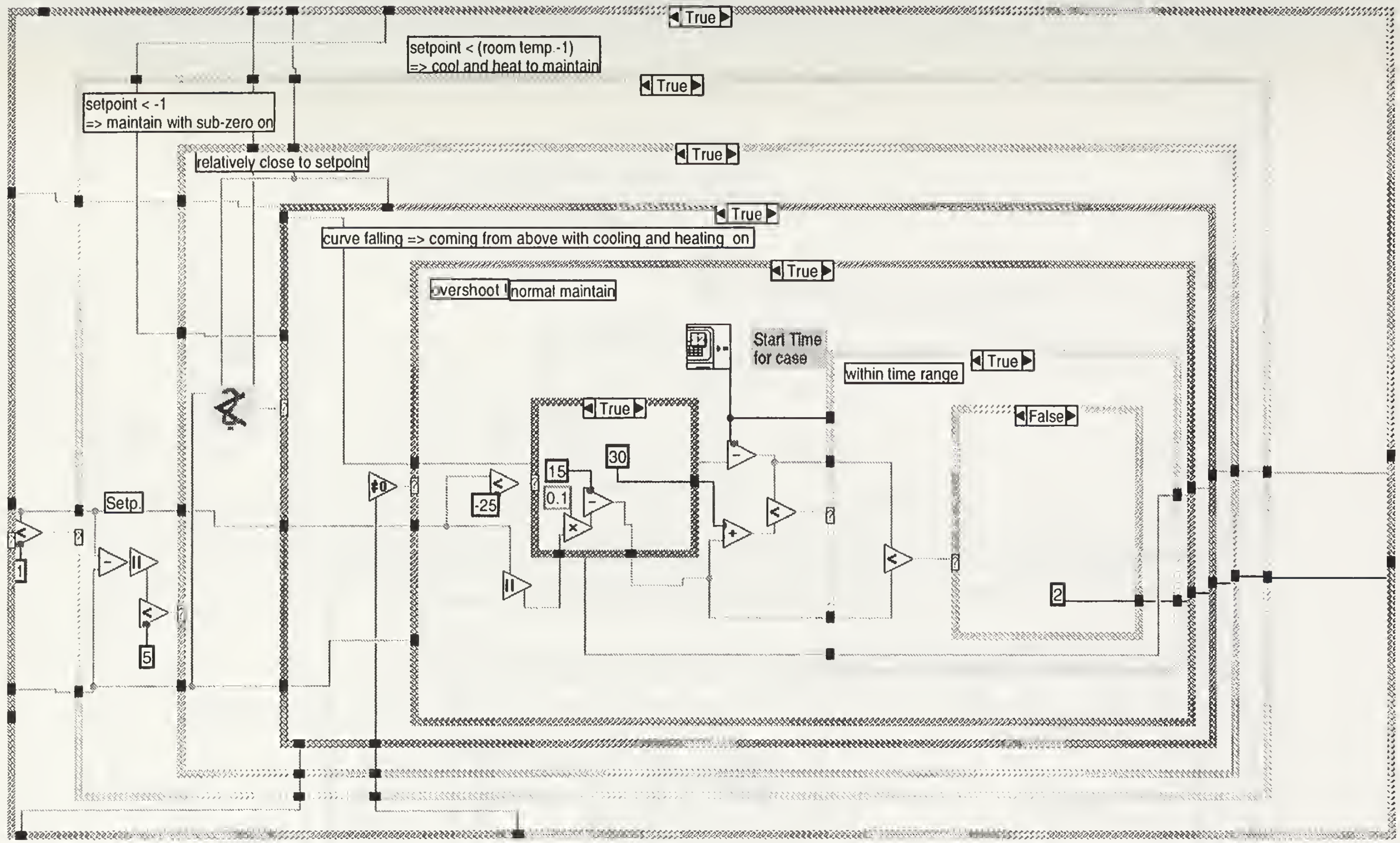

True
within time range
True
True
False
0
15
15
6
False
0
True
True
False
0
15
6
False
56

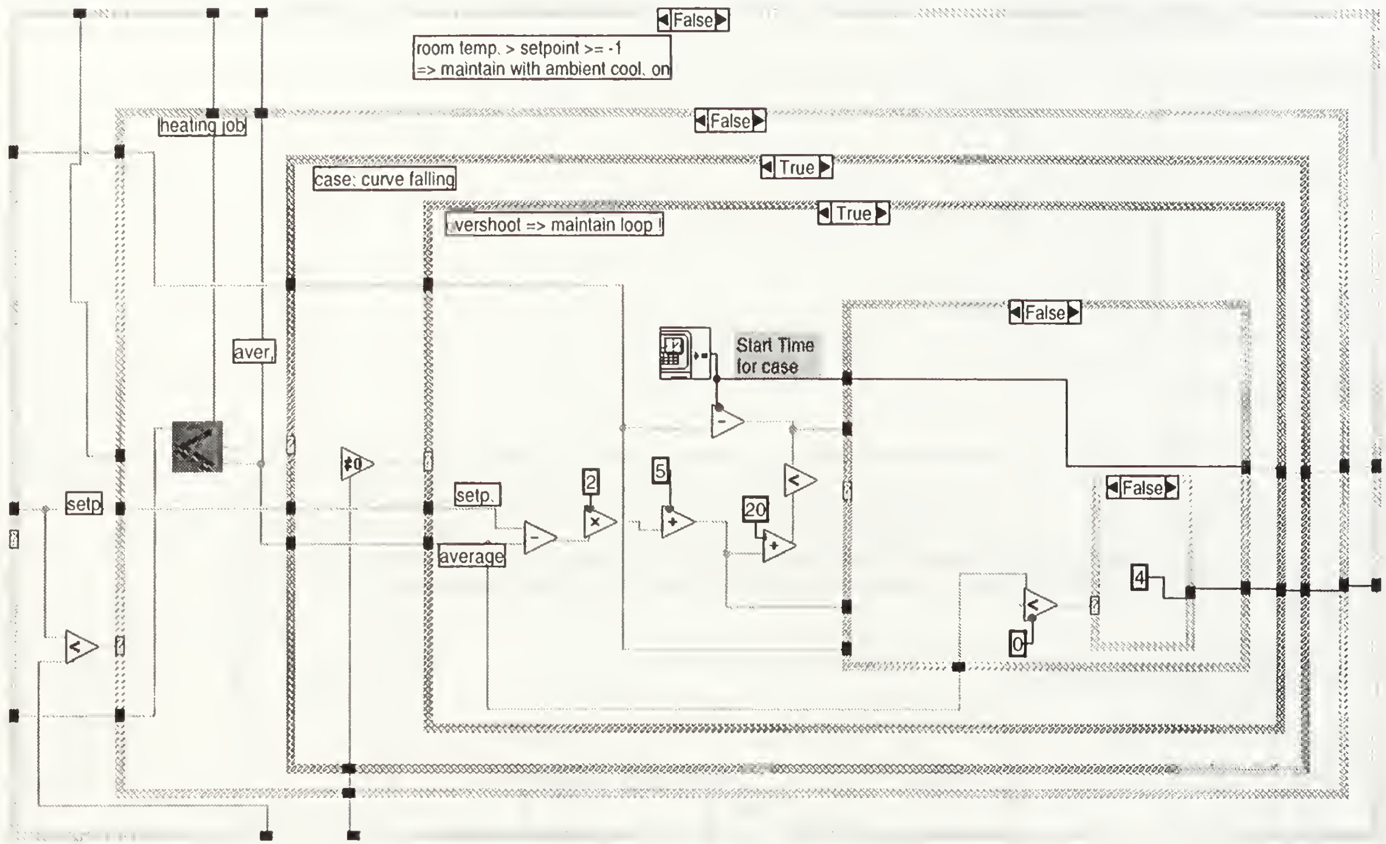

True
False
0
False
0
True
56
True
no overshoot
False
TVAC rising
case: r.o.c sinks
Start Time
for case
True
within time range
False
0
True
setp.
average
1
30
5

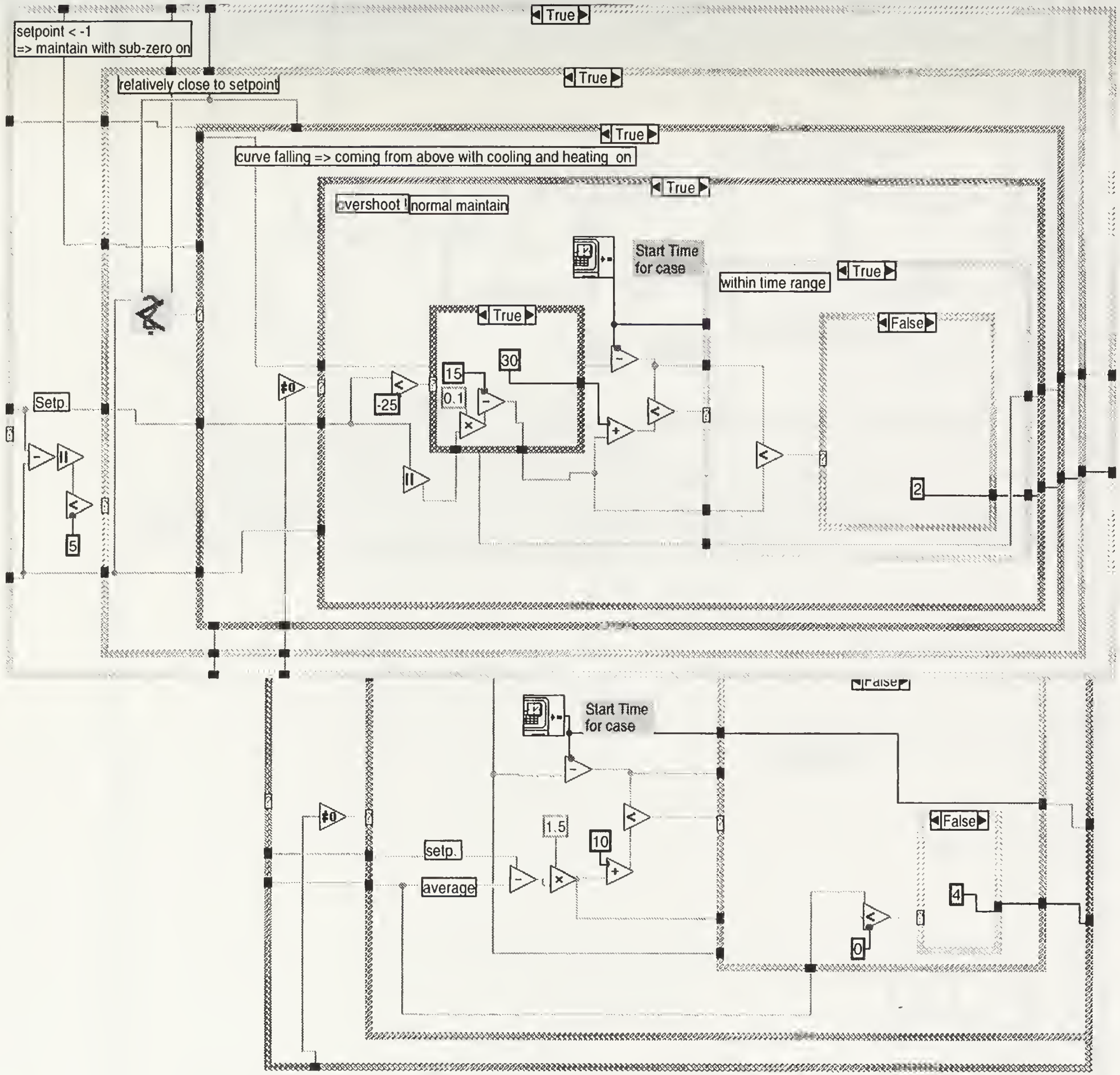

False
TVAC rising
case: r.o.c sinks
Start Time
for case
True
within time range
False
True
setp.
1
30
average
5
0
False
2.2
True
30
False
0
True
within time range
False
0
False
0

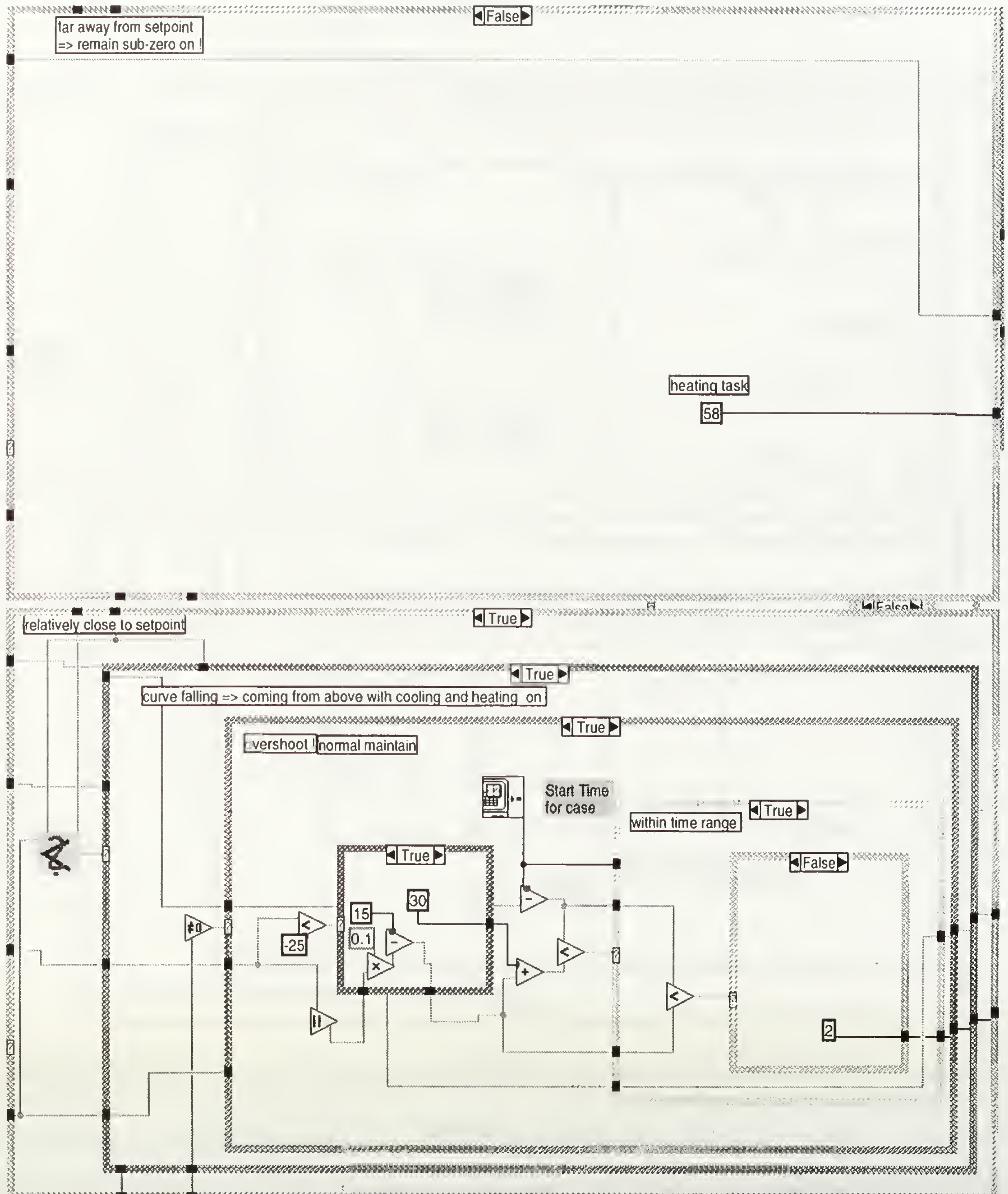

True
56
True
TVAC temp is falling
within time range
True
Start Time
for case
True
56
5
setp.
5
False
0

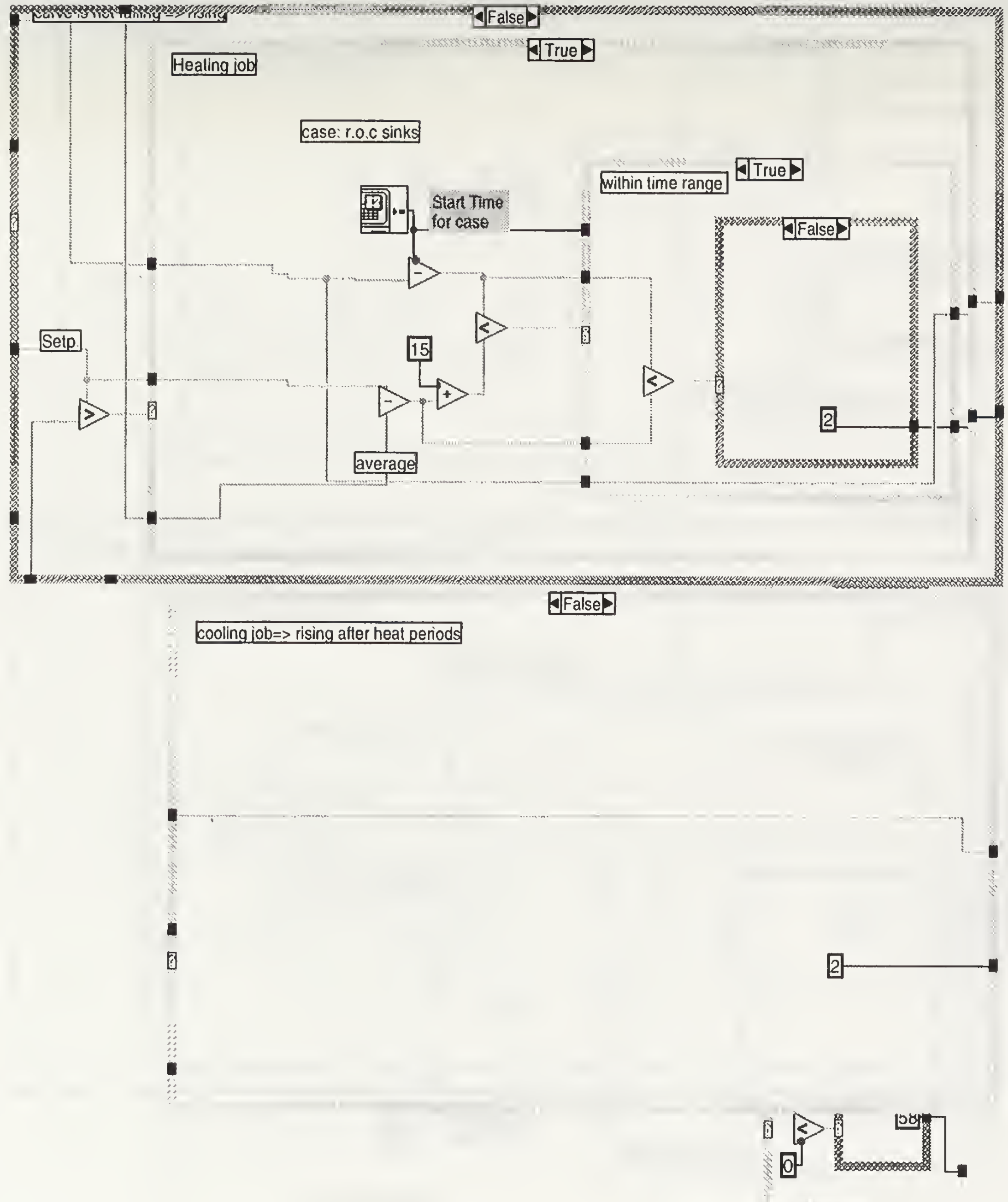

within time range
True
True
56
False
0
True
56

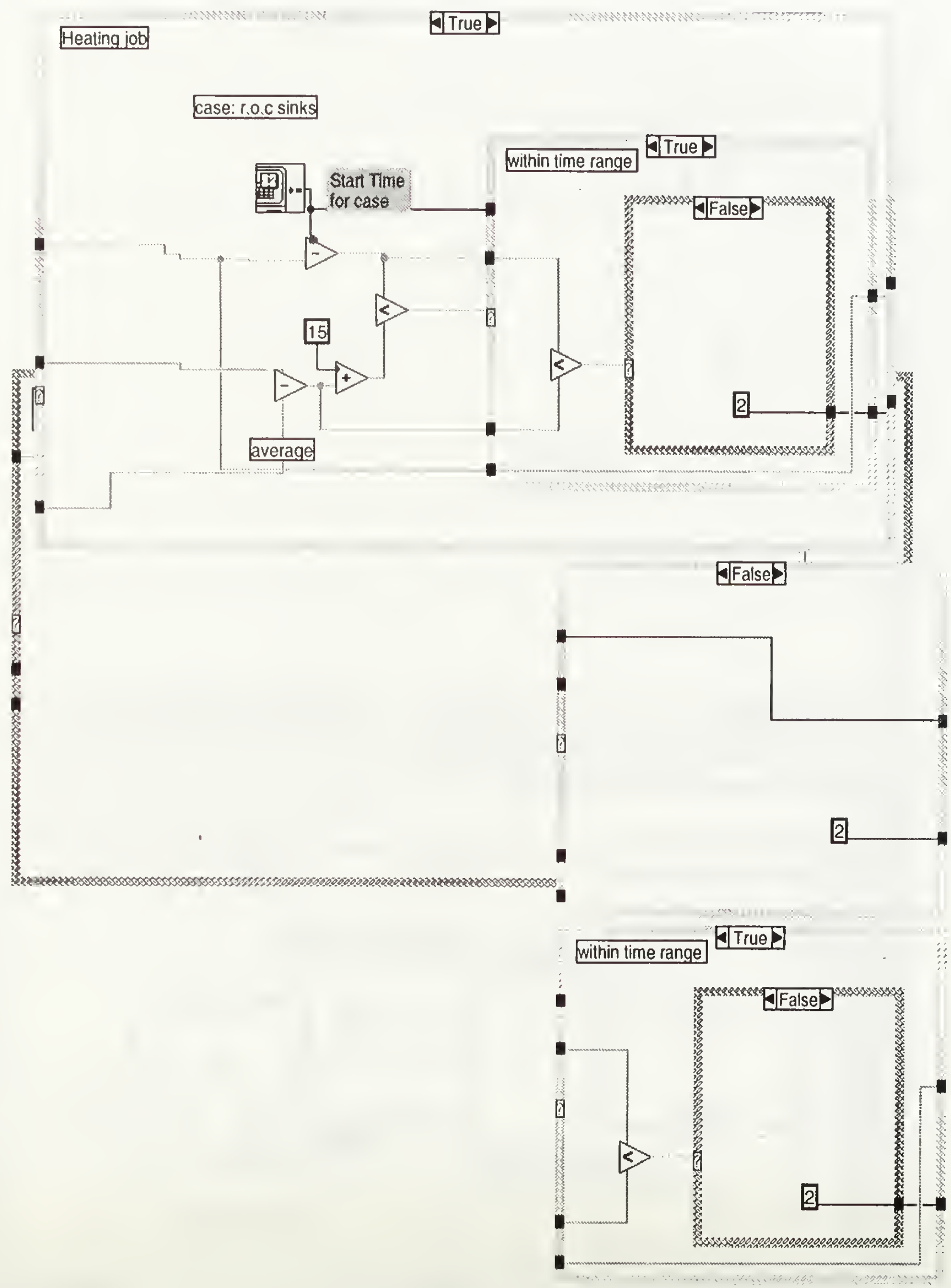

True
setpoint < (room temp.-1)
=> cool and heat to maintain
True
setpoint < -1
=> maintain with sub-zero on
True
relatively close to setpoint
True
curve falling => coming from above with cooling and heating on
True
overshoot !
normal maintain
Start Time
for case
within time range
True
False
True
30
15
0.1
-25
2
Setp.
5
1

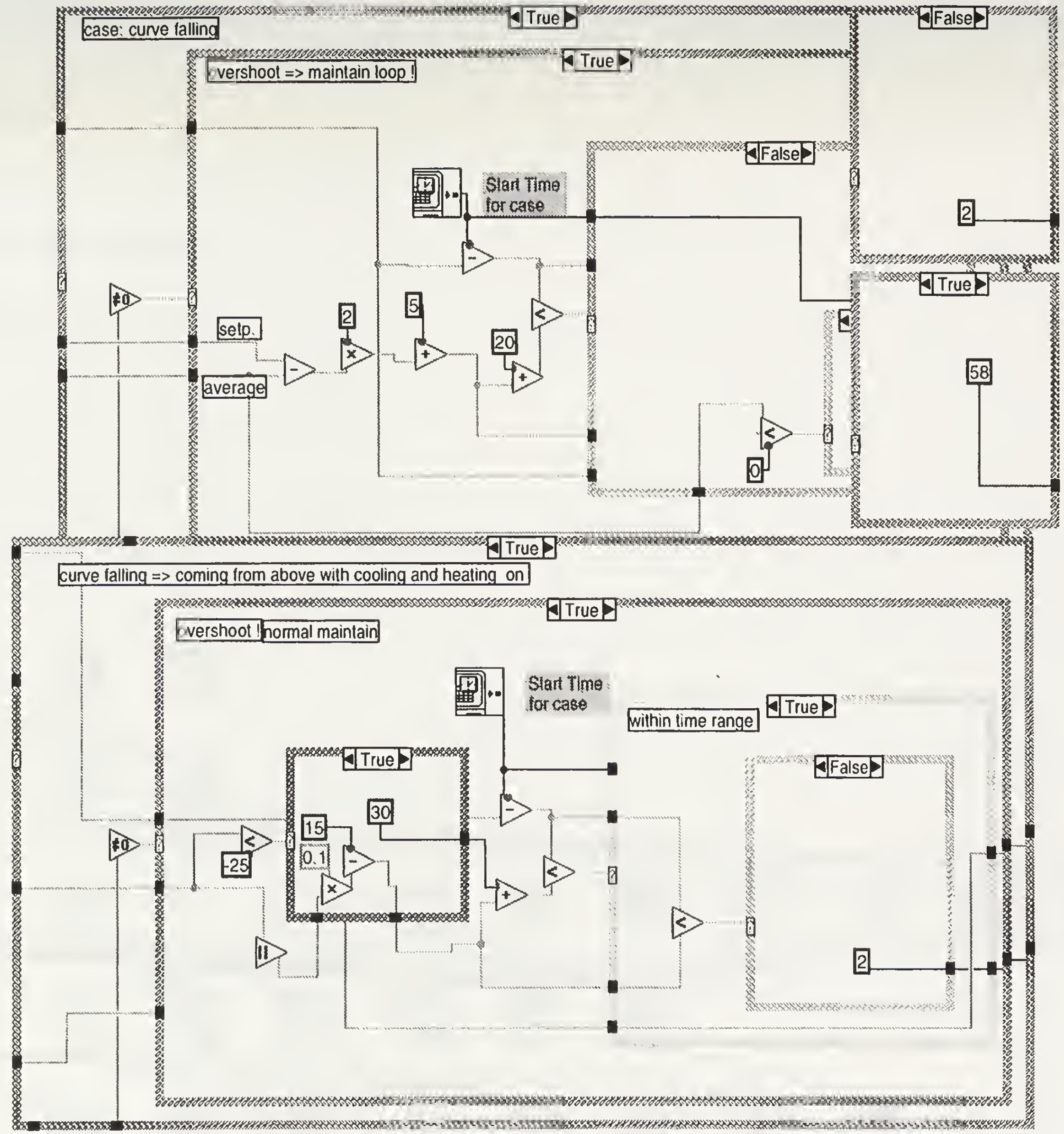

False
room temp. > setpoint >= -1
=> maintain with ambient cool. on
heating job
False
case: curve falling
True
vershoot => maintain loop !
True
aver.
Start Time
for case
False
setp
setp.
average
2
5
20
False
4
0


True
setpoint < -1
=> maintain with sub-zero on
True
relatively close to setpoint
True
curve falling => coming from above with cooling and heating on
True
overshoot !
normal maintain
Start Time
for case
within time range
True
True
False
15
30
0.1
-25
2
Setp.
5
False
Start Time
for case
1.5
10
setp.
average
False
4
0

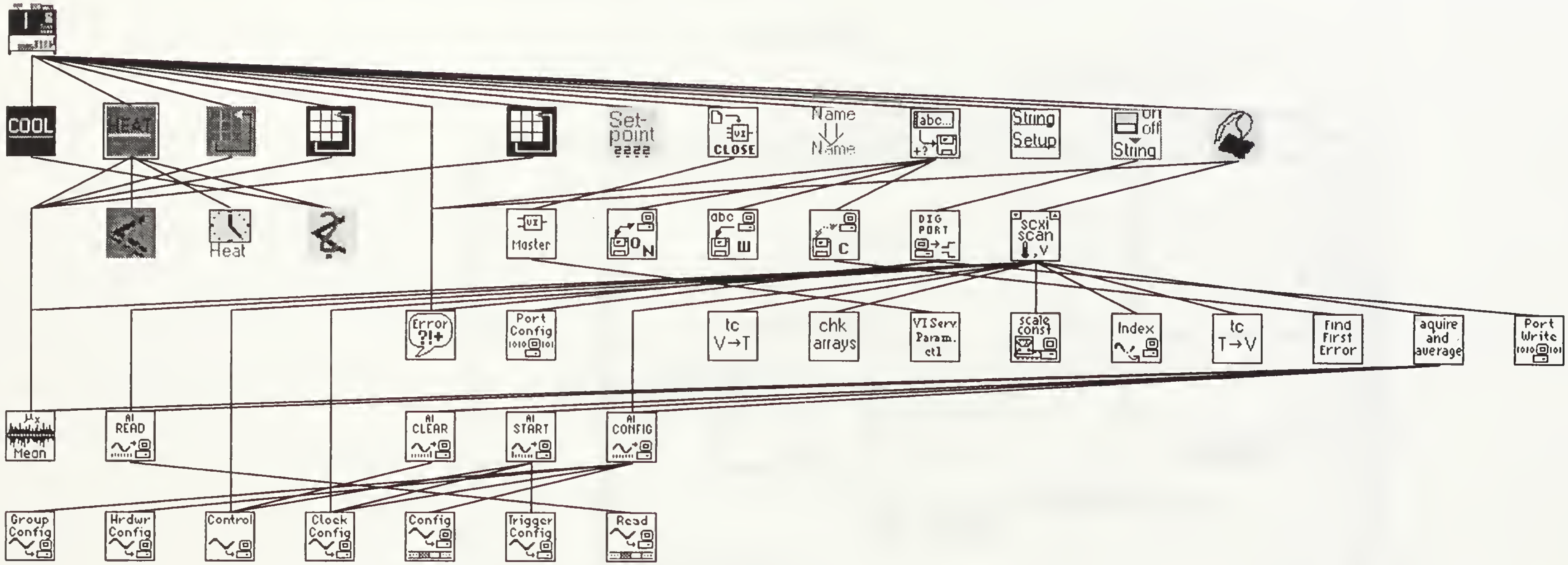

False
far away from setpoint
=> remain sub-zero on !
heating task
58
True
relatively close to setpoint
True
curve falling => coming from above with cooling and heating on
True
overshoot !
normal maintain
Start Time
for case
within time range
True
True
False
15
30
0.1
-25
2


False
True
Heating job
case: r.o.c sinks
within time range
True
Start Time
for case
False
Setp
15
2
average
False
cooling job=> rising after heat periods
2
0
58


True
Heating job
case: r.o.c sinks
Start Time
for case
within time range
True
False
15
average
2
False
2
within time range
True
False
2


True
case: curve falling
True
overshoot => maintain loop !
False
Start Time
for case
setp.
average
2
5
20
0
False
2
True
58
True
curve falling => coming from above with cooling and heating on
True
overshoot ! normal maintain
Start Time
for case
within time range
True
True
30
15
0.1
-25
False
2

Position in Hierarchy



List of SubVIs

falling2.vi
C:\LABVIEW\EXAMPLES\DAQ\TVACUUM.LLB\falling2.vi

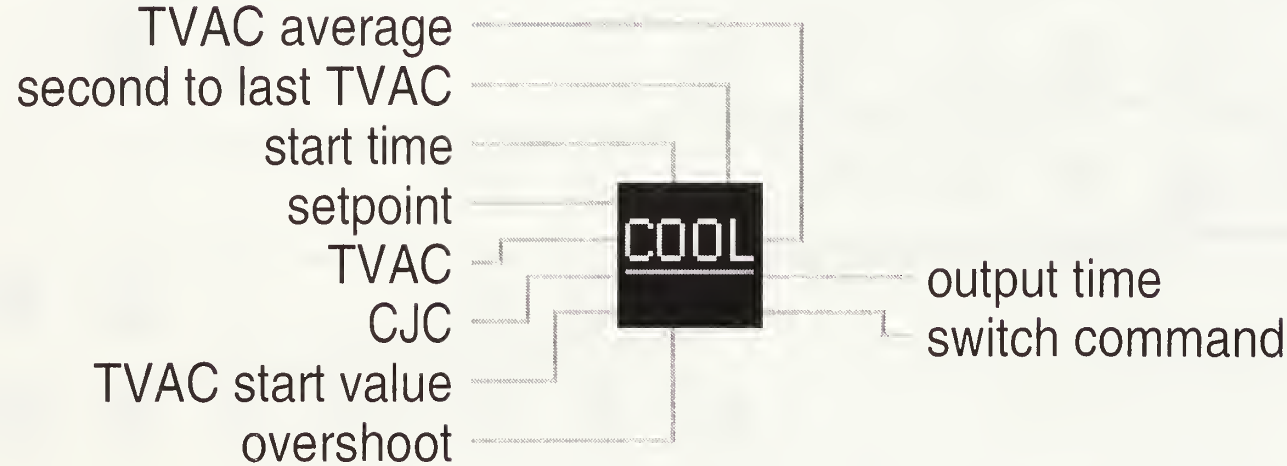
Mean.vi
C:\LABVIEW\vi.lib\ANALYSIS\5STAT.LLB\Mean.vi

determine full heating time.vi
C:\LABVIEW\EXAMPLES\DAQ\TVACUUM.LLB\determine full heating time.vi

falling or constant.vi
C:\LABVIEW\EXAMPLES\DAQ\TVACUUM.LLB\falling or constant.vi

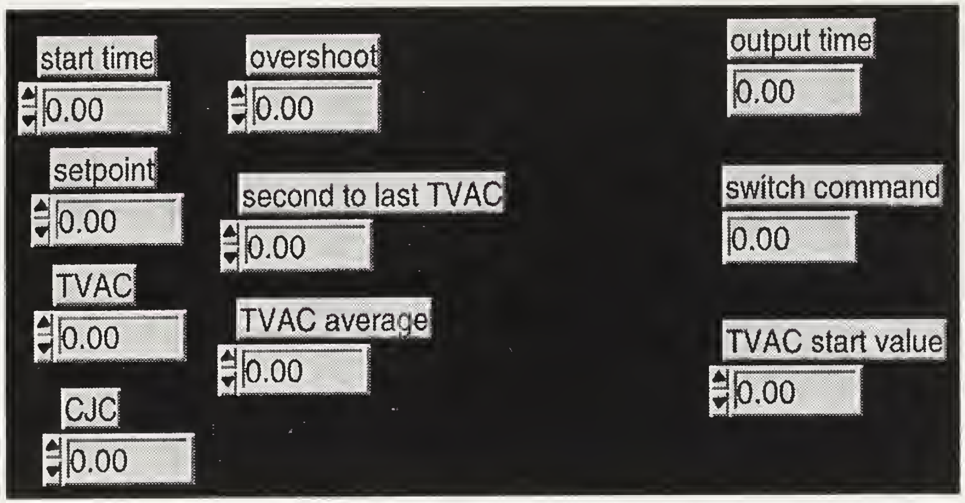
APPENDIX H
DOCUMENTATION OF COOLING.VI

Connector Pane



Cooling.vi

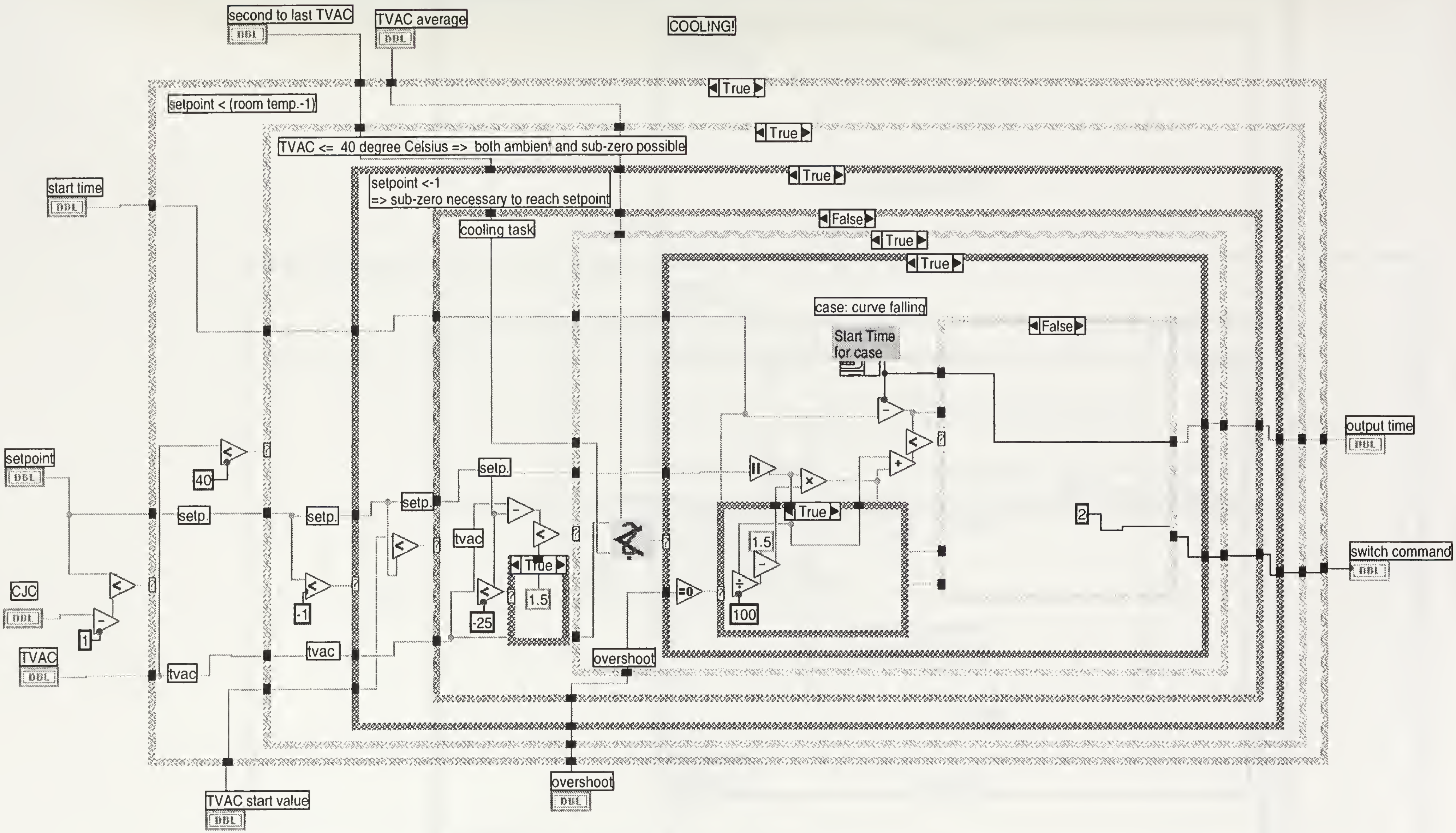
Front Panel

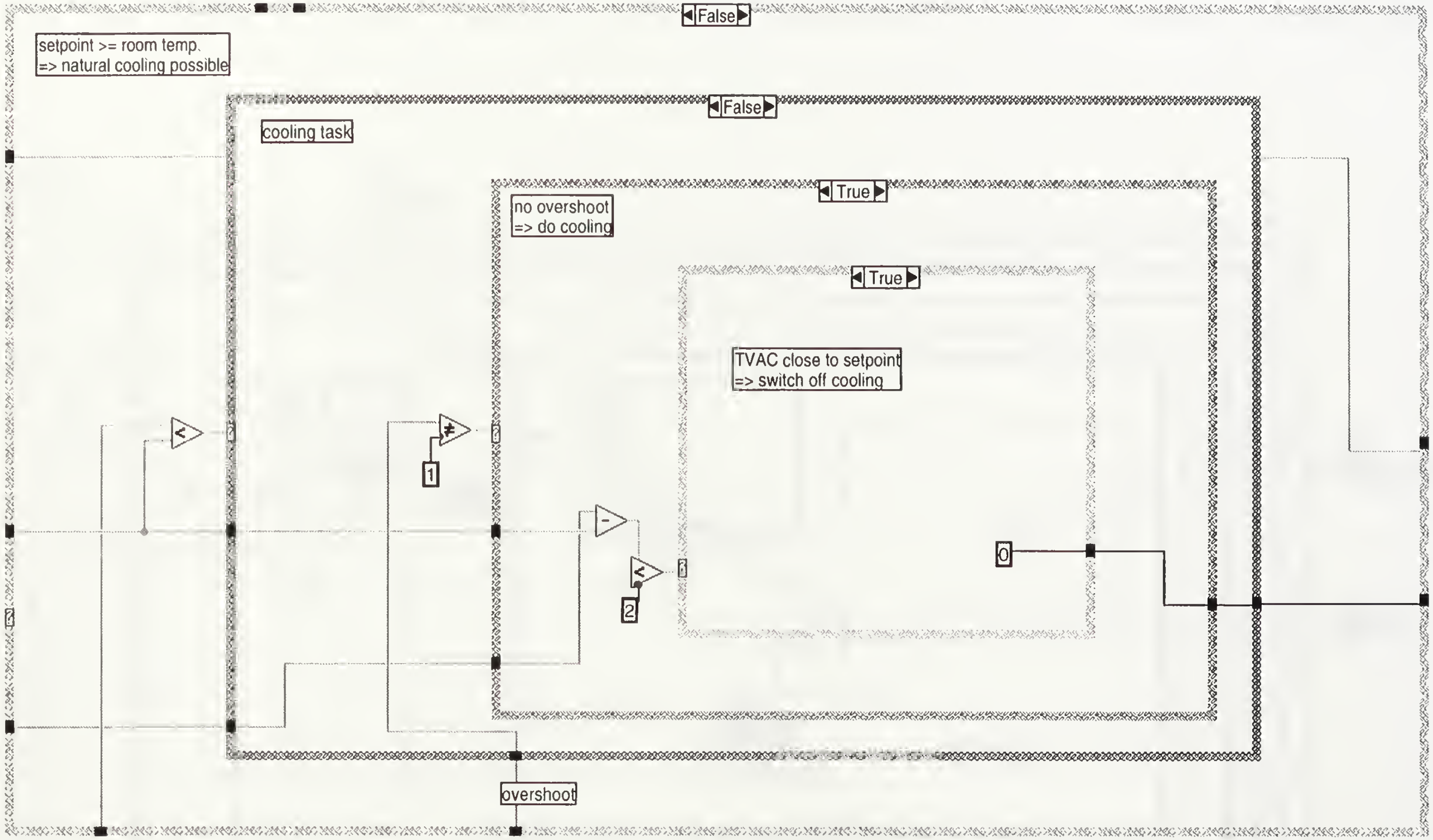

start time
0.00
setpoint
0.00
TVAC
0.00
CJC
0.00
overshoot
0.00
second to last TVAC
0.00
TVAC average
0.00
output time
0.00
switch command
0.00
TVAC start value
0.00

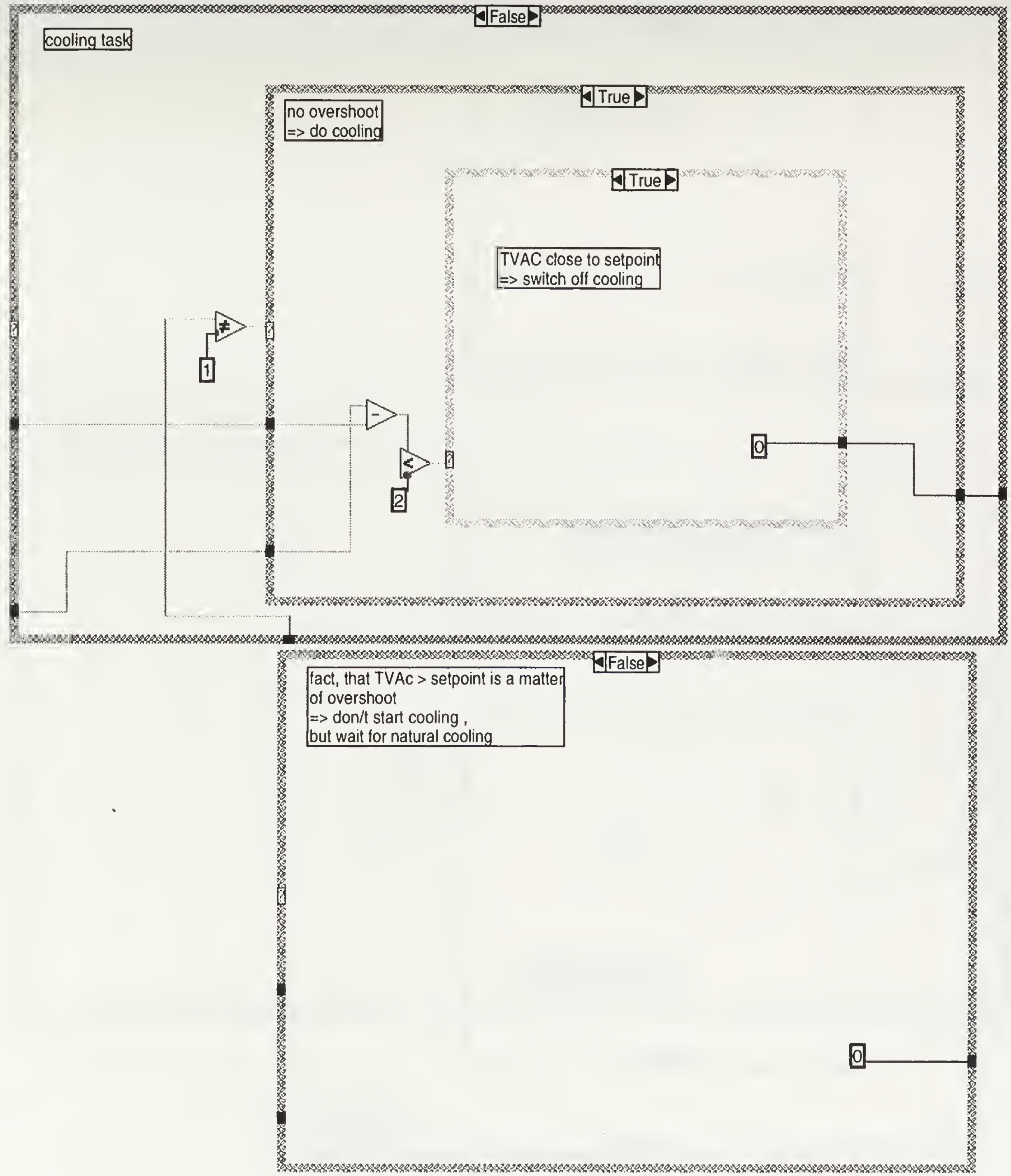
Controls and Indicators

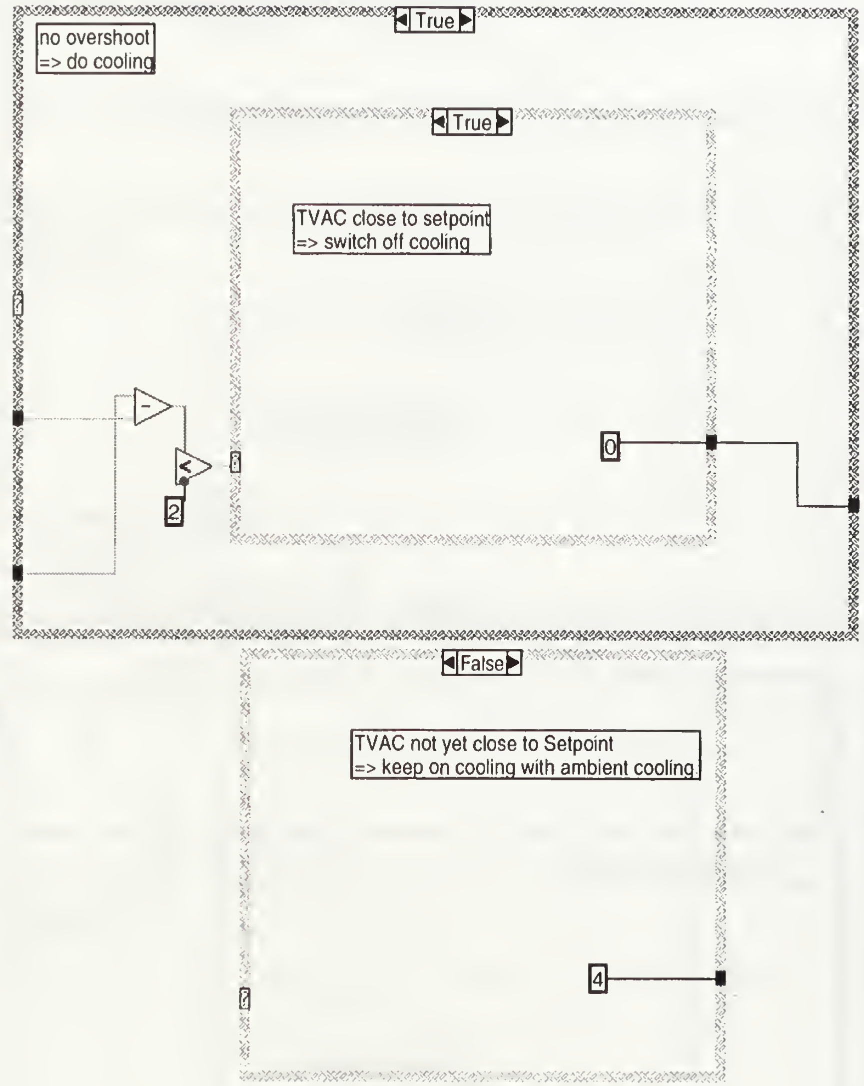
- DBL **start time**
- DBL **setpoint**
- DBL **TVAC**
- DBL **CJC**
- DBL **overshoot**
- DBL **second to last TVAC**
- DBL **TVAC average**
- DBL **TVAC start value**
- DBL **output time**
- DBL **switch command**

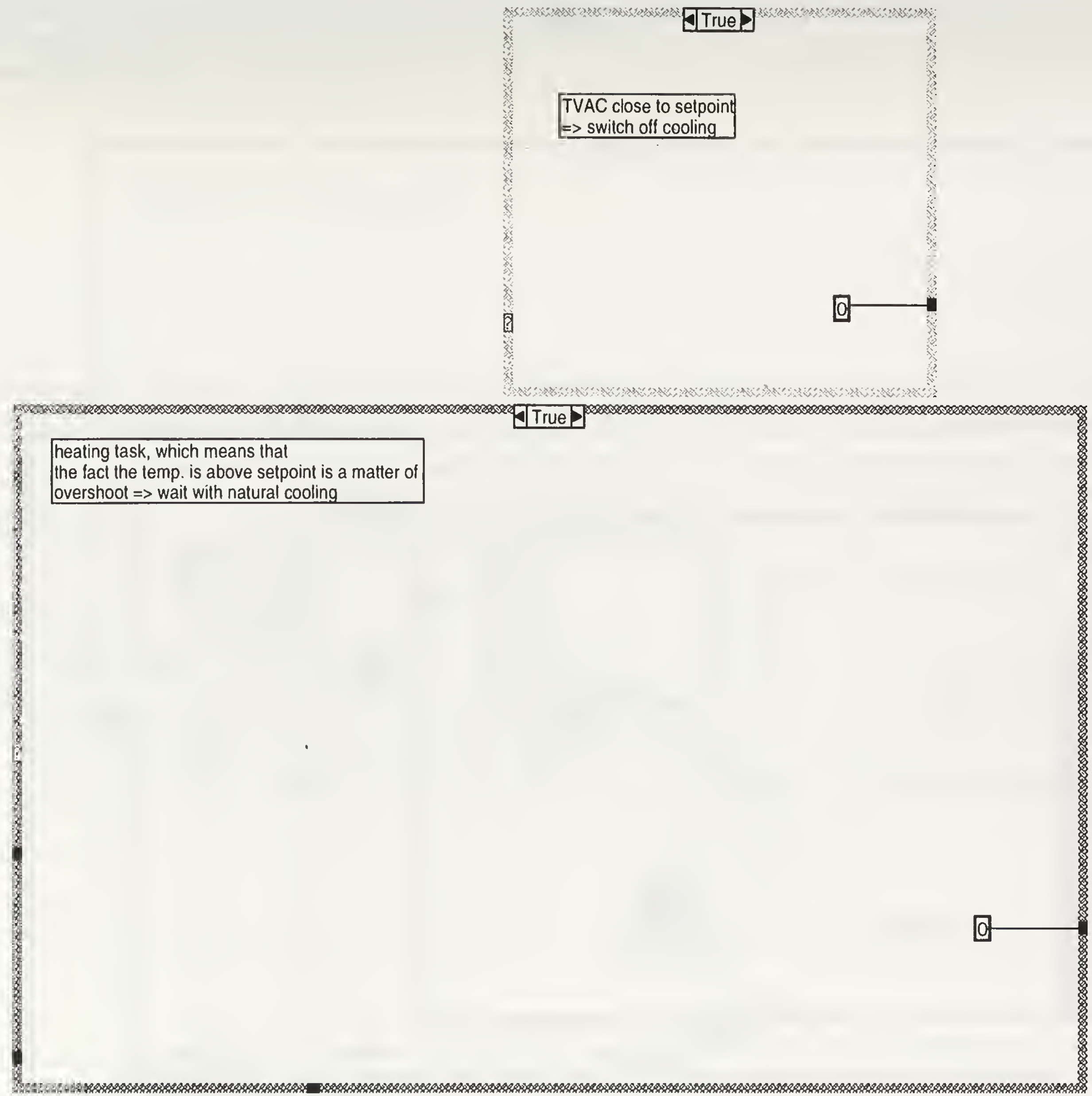
Block Diagram

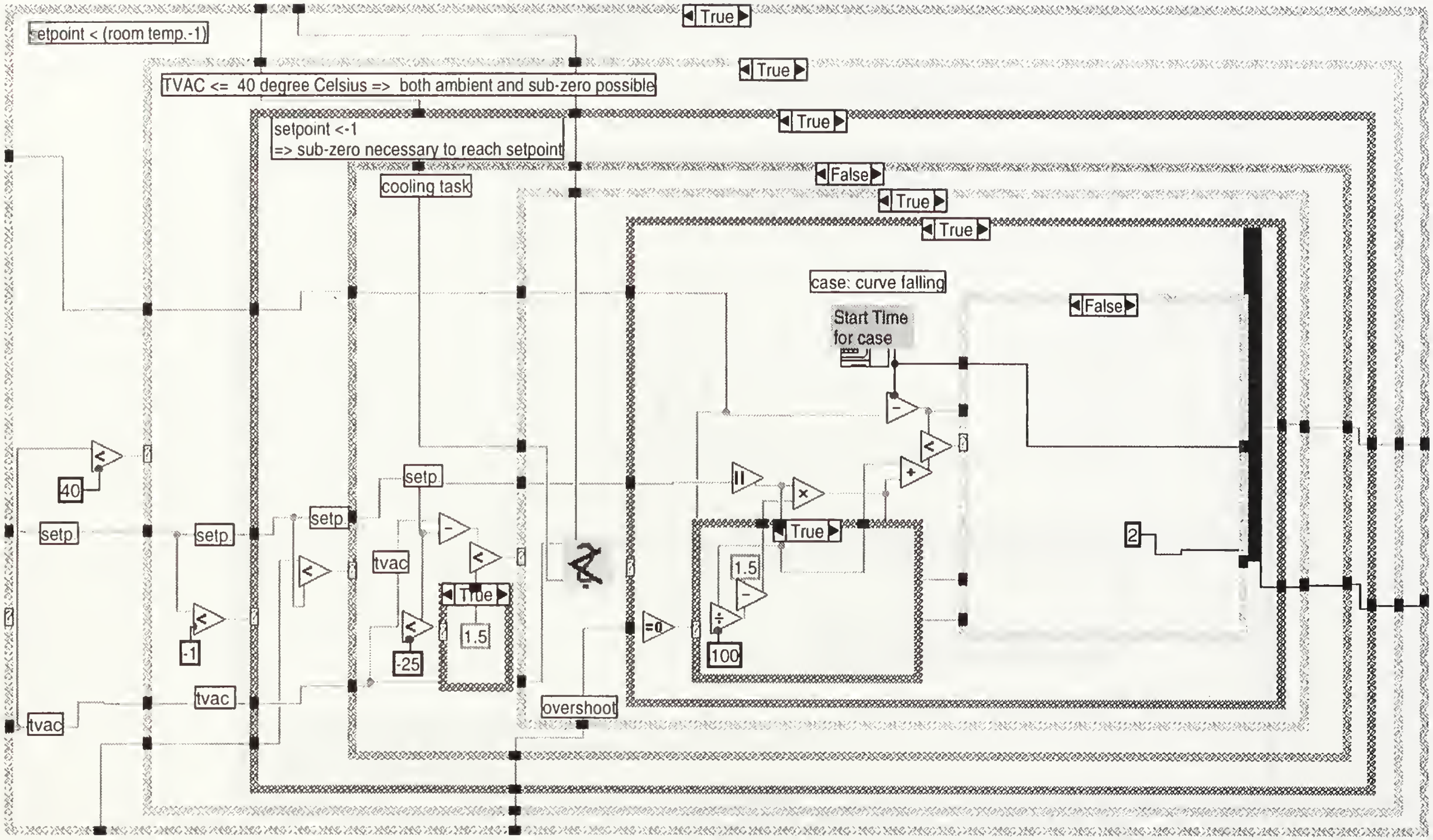


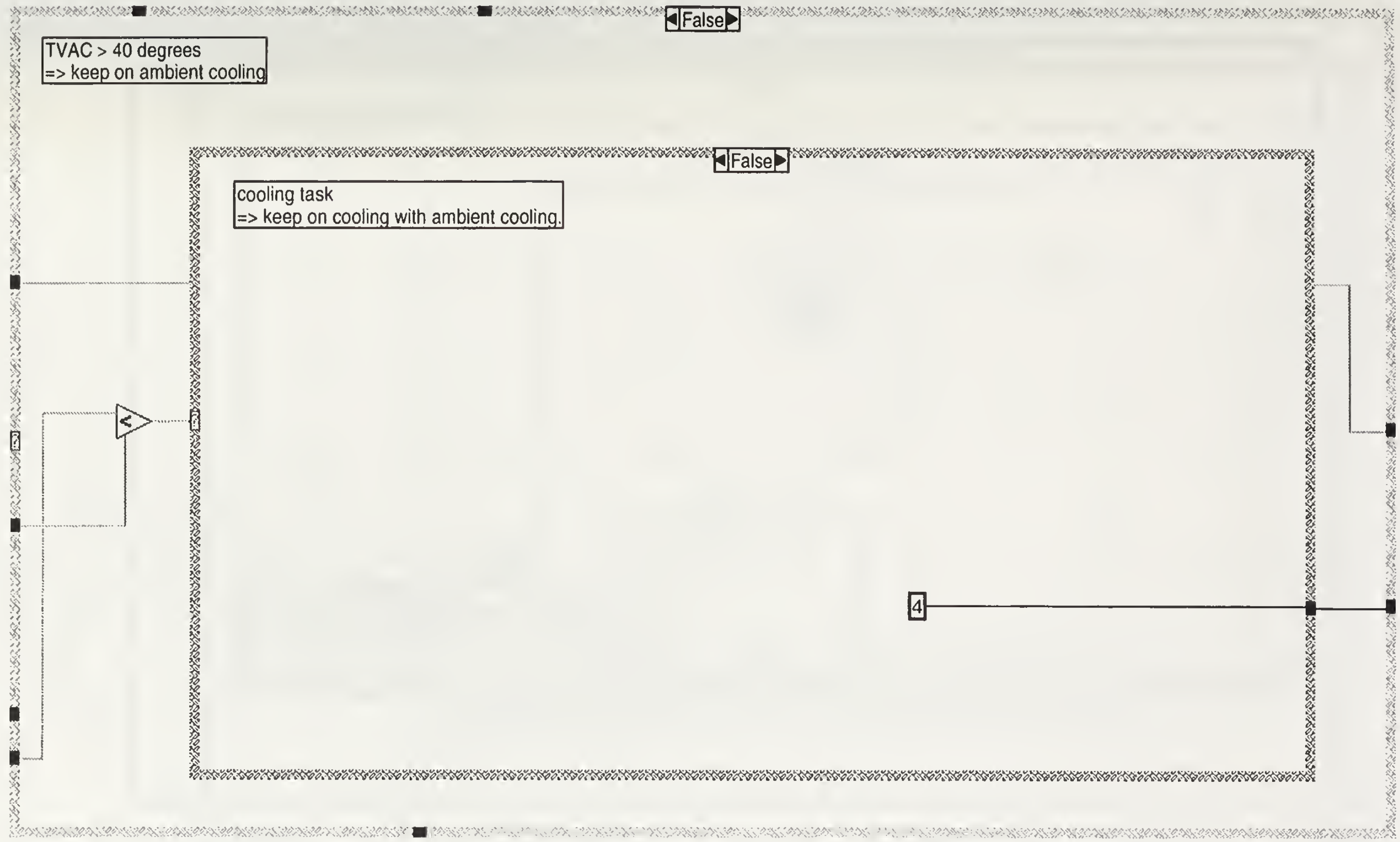

False
setpoint >= room temp.
=> natural cooling possible
False
cooling task
True
no overshoot
=> do cooling
True
TVAC close to setpoint
=> switch off cooling
1
2
0
overshoot

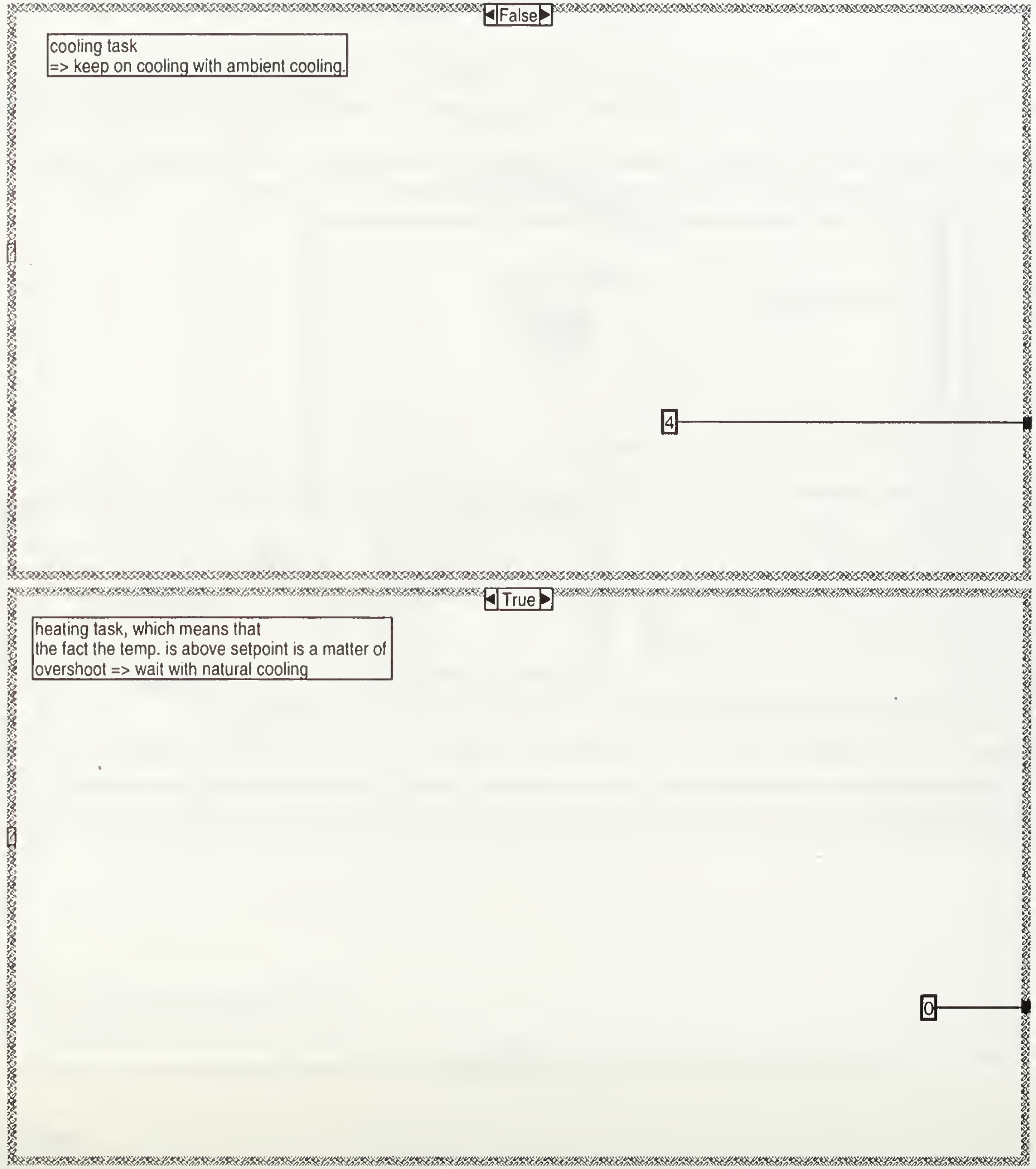

False
cooling task
True
no overshoot
=> do cooling
True
TVAC close to setpoint
=> switch off cooling
1
2
0
False
fact, that TVAc > setpoint is a matter
of overshoot
=> don/t start cooling ,
but wait for natural cooling
0

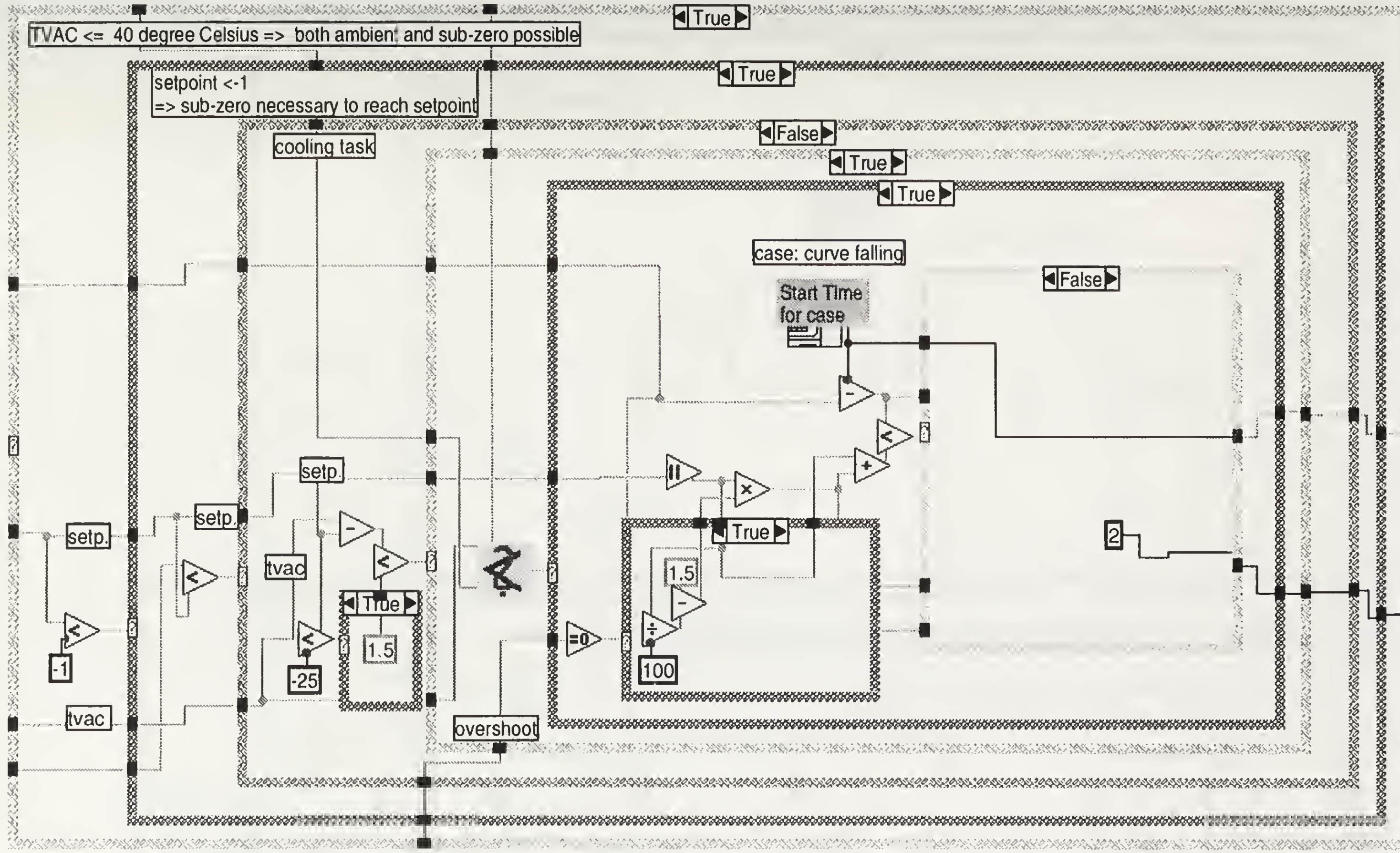

True
no overshoot
=> do cooling
True
TVAC close to setpoint
=> switch off cooling
2
0
False
TVAC not yet close to Setpoint
=> keep on cooling with ambient cooling
4

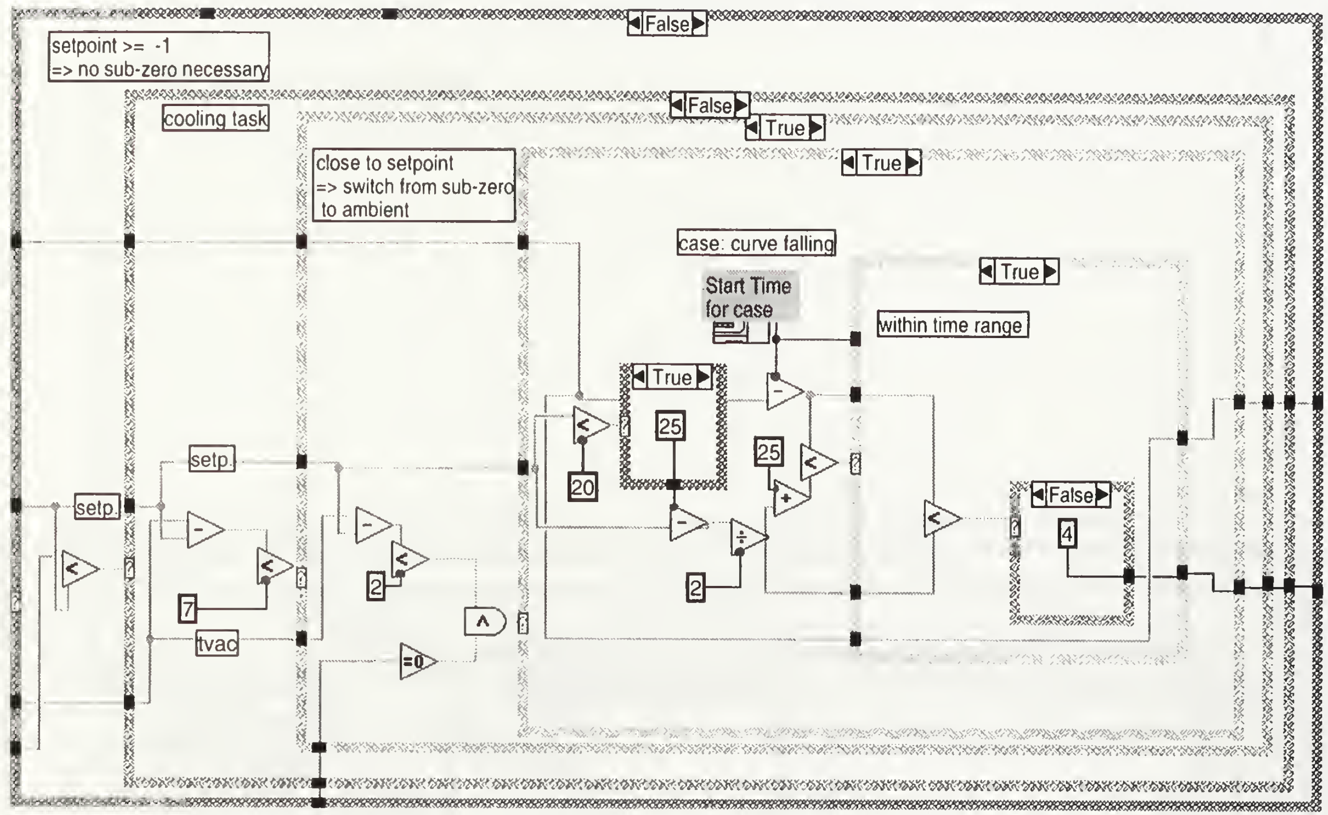

True
TVAC close to setpoint
=> switch off cooling
0
True
heating task, which means that
the fact the temp. is above setpoint is a matter of
overshoot => wait with natural cooling
0

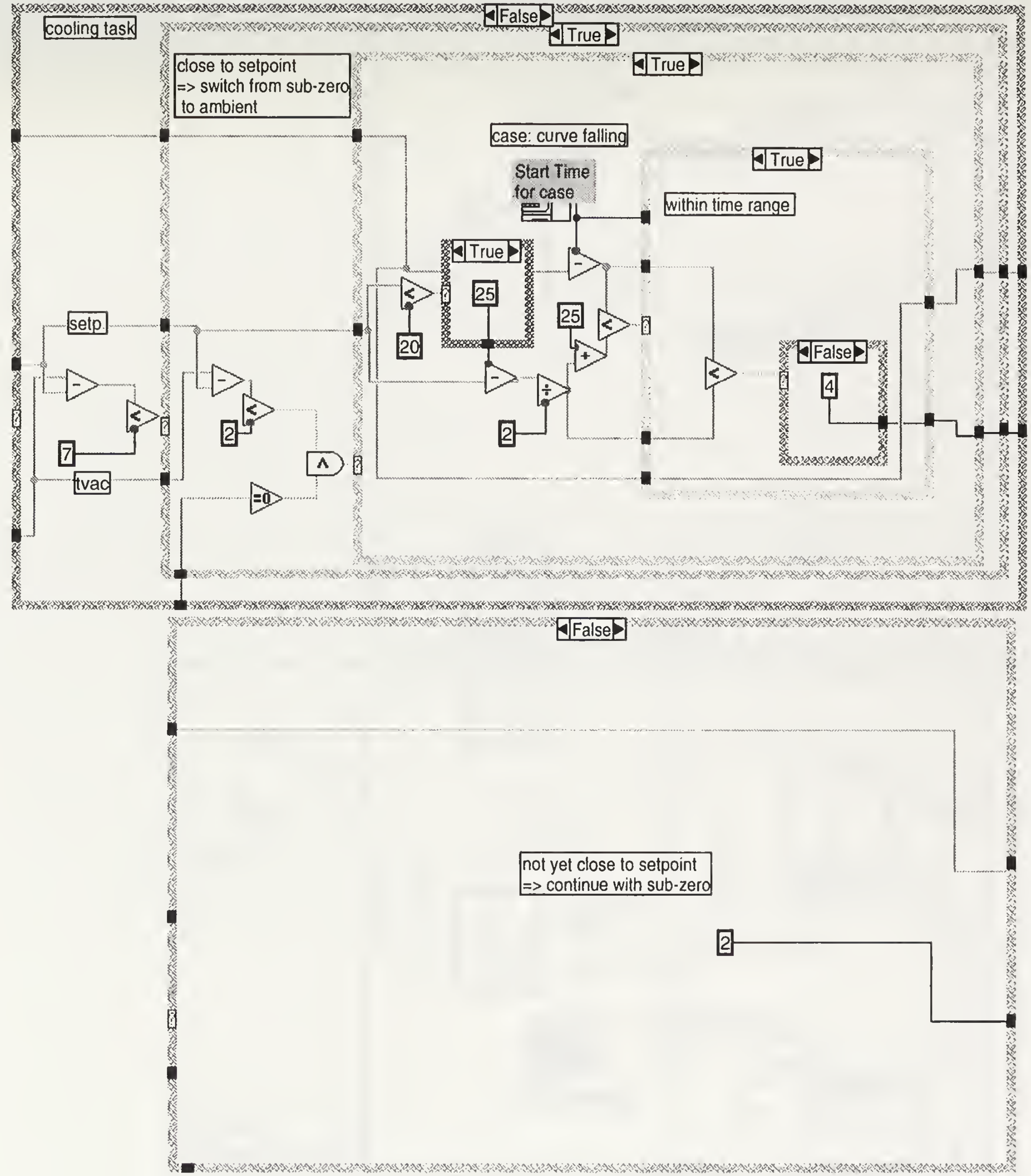

True
setpoint < (room temp.-1)
True
TVAC <= 40 degree Celsius => both ambient and sub-zero possible
True
setpoint <-1
=> sub-zero necessary to reach setpoint
False
cooling task
True
True
case: curve falling
False
Start Time
for case
40
setp
setp
setp
setp
-1
tvac
tvac
tvac
-25
True
1.5
overshoot
=0
True
1.5
100
2

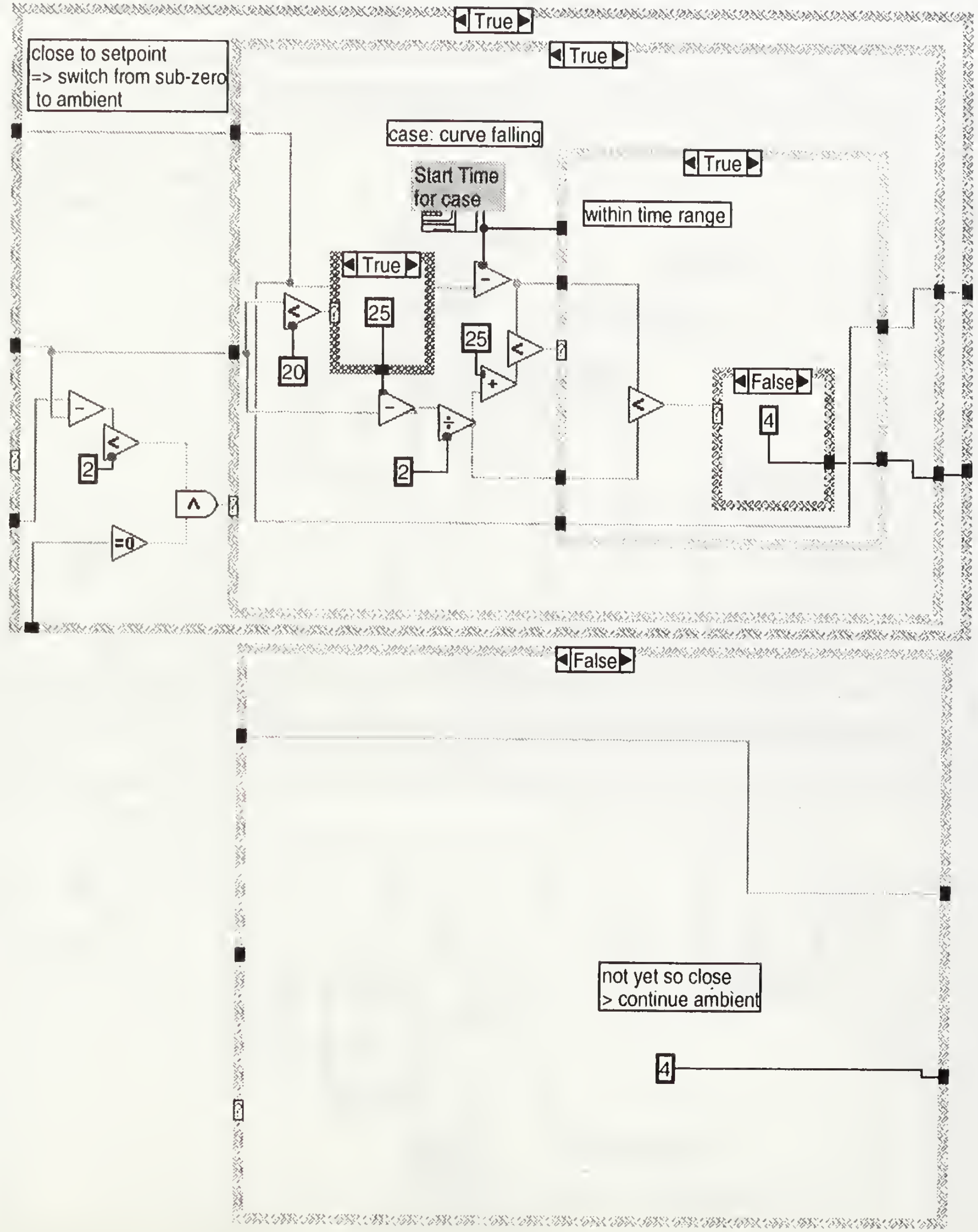

False
TVAC > 40 degrees
=> keep on ambient cooling
False
cooling task
=> keep on cooling with ambient cooling.
4

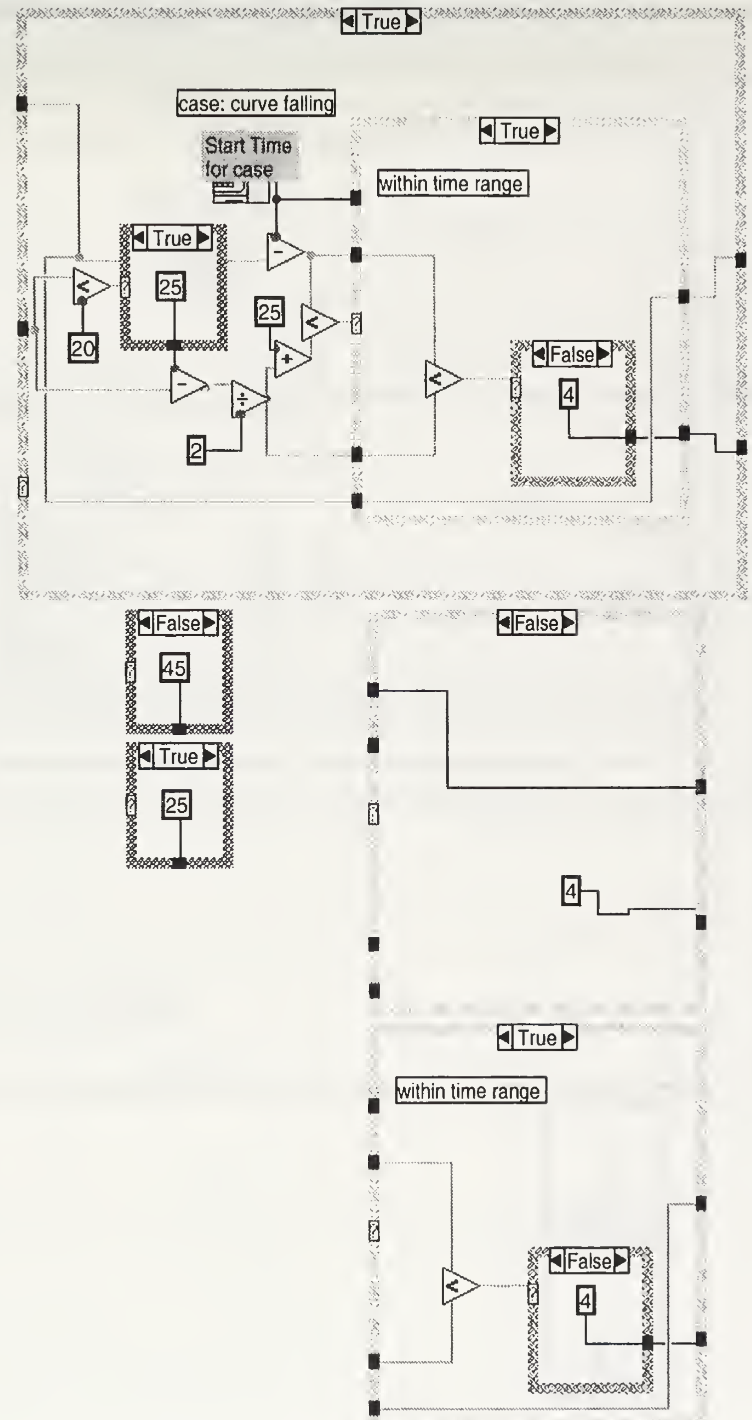

False
cooling task
=> keep on cooling with ambient cooling.
4
True
heating task, which means that
the fact the temp. is above setpoint is a matter of
overshoot => wait with natural cooling
0

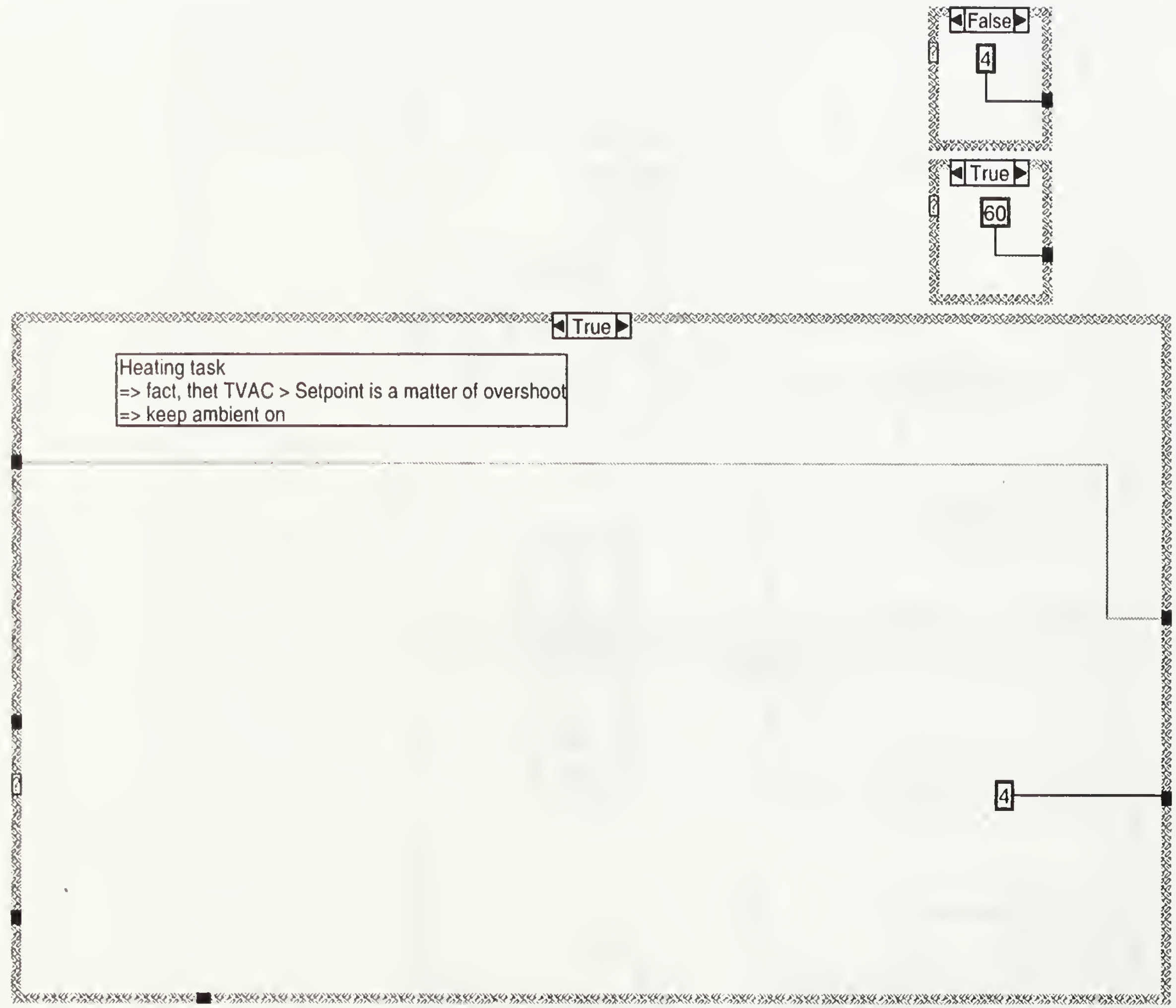

TVAC <= 40 degree Celsius => both ambient and sub-zero possible
True
setpoint <-1
=> sub-zero necessary to reach setpoint
True
cooling task
False
True
True
case: curve falling
Start Time
for case
False
setp.
-1
tvac
setp.
setp.
tvac
-25
True
1.5
overshoot
=0
100
1.5
True
2

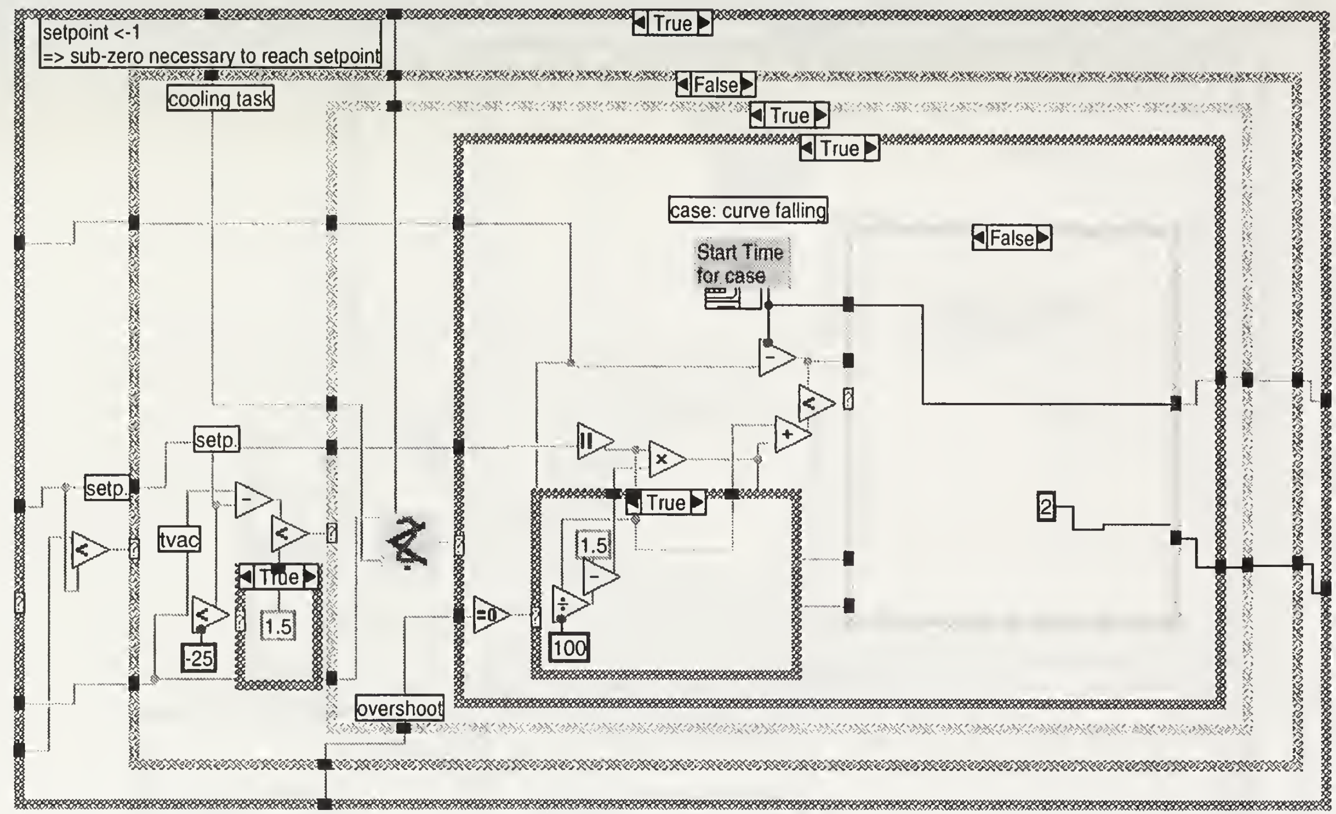

COOL
False
setpoint >= -1
=> no sub-zero necessary
False
cooling task
True
close to setpoint
=> switch from sub-zero
to ambient
True
case: curve falling
True
Start Time
for case
within time range
True
25
25
20
setp.
setp.
False
4
7
2
2
tvac
=0

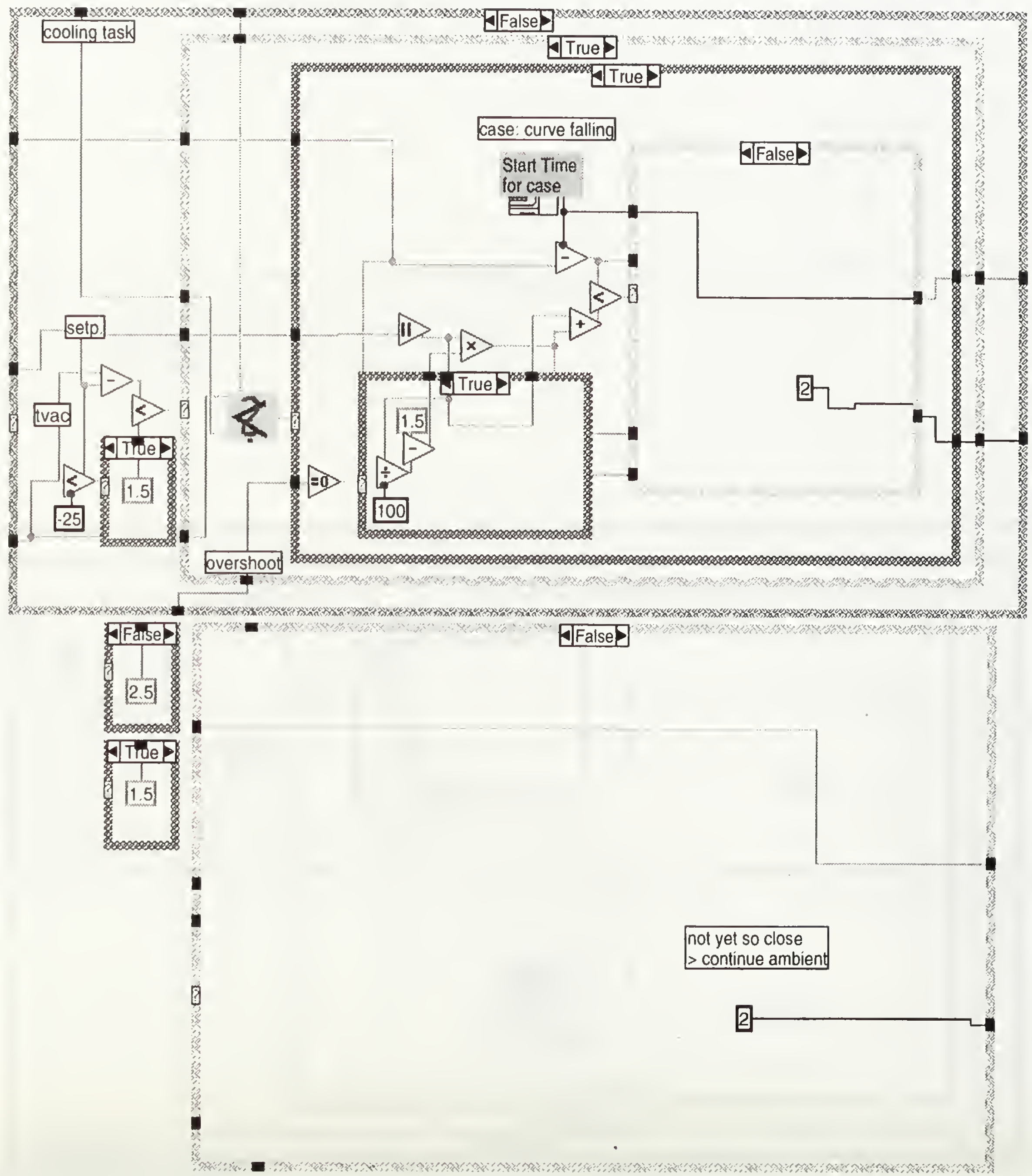

False
cooling task
True
True
close to setpoint
=> switch from sub-zero
to ambient
case: curve falling
True
Start Time
for case
within time range
True
25
20
25
2
False
4
setp
7
tvac
2
=0
False
not yet close to setpoint
=> continue with sub-zero
2

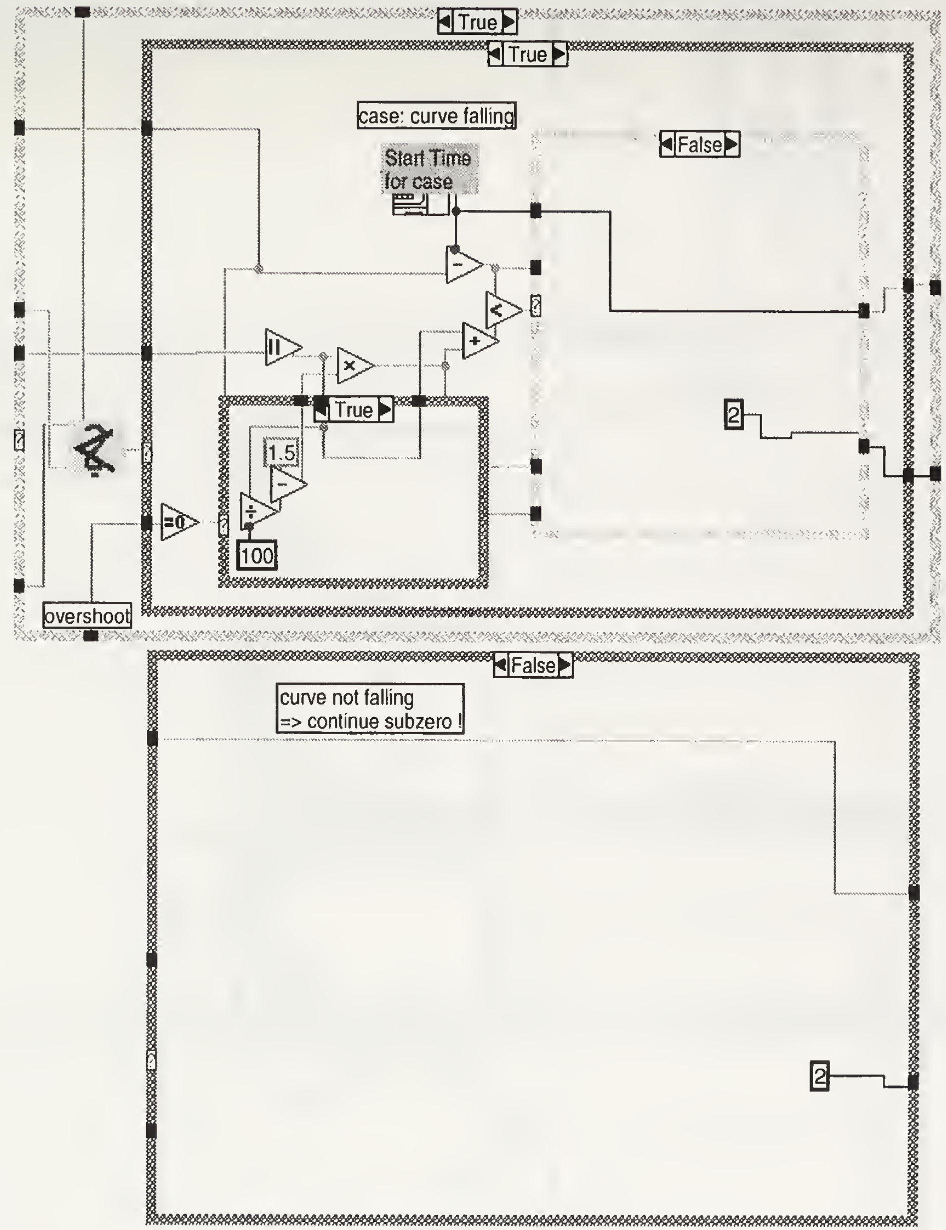

True
close to setpoint
=> switch from sub-zero
to ambient
True
case: curve falling
Start Time
for case
True
within time range
True
25
20
25
2
False
4
2
=0
False
not yet so close
> continue ambient
4

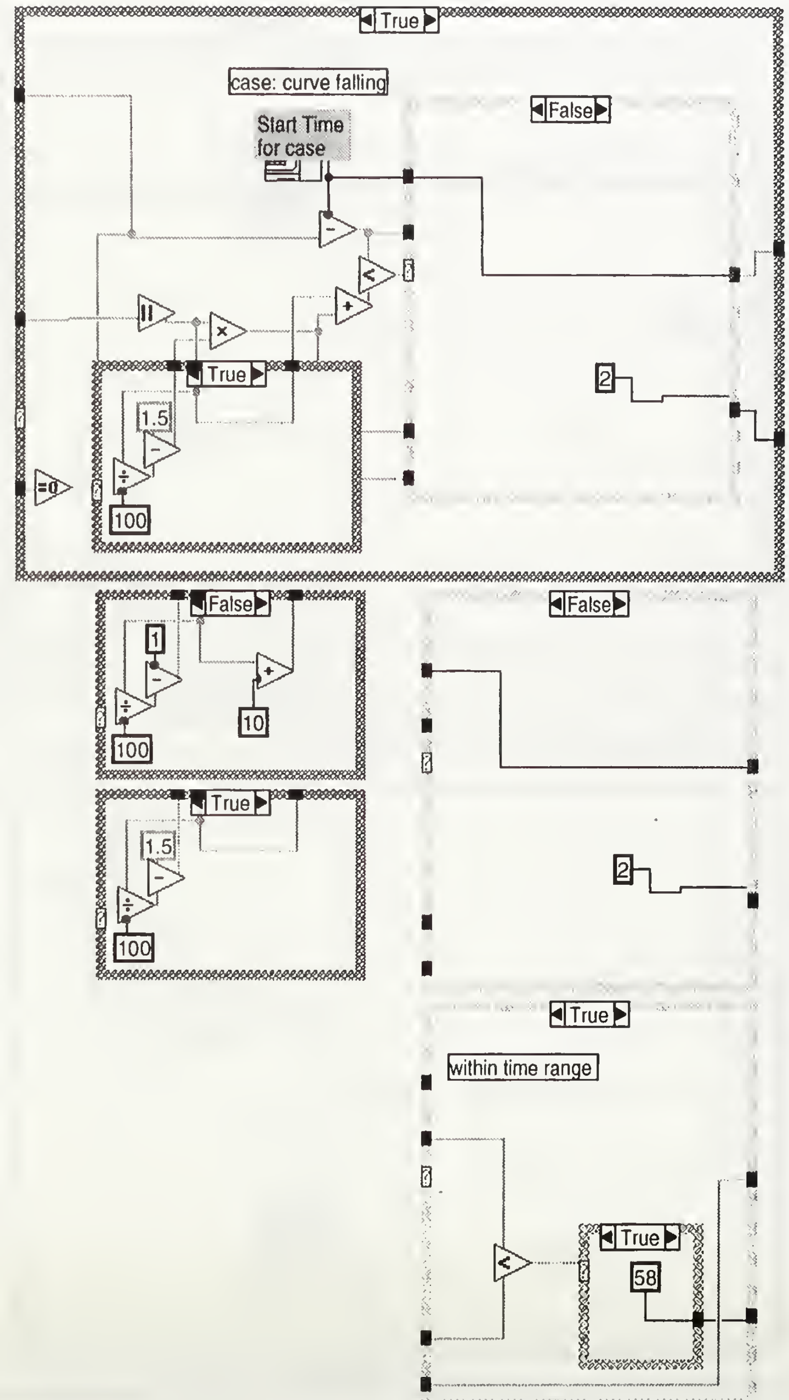

True
case: curve falling
Start Time for case
True
within time range
True
25
20
25
2
False
4
False
45
True
25
False
4
True
within time range
False
4

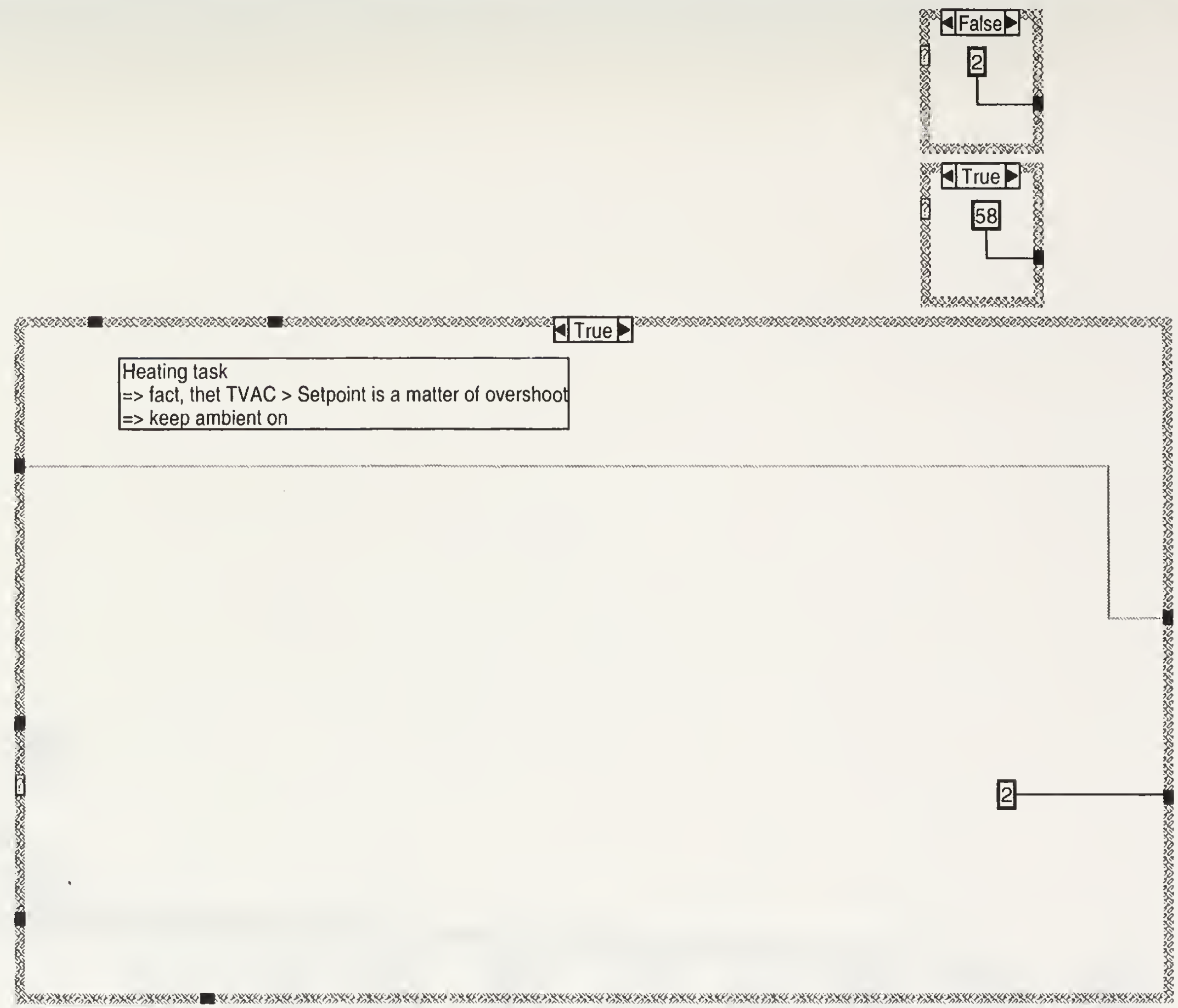

COOL
False
4
True
60
True
Heating task
=> fact, thet TVAC > Setpoint is a matter of overshoot
=> keep ambient on
4


setpoint <-1
=> sub-zero necessary to reach setpoint
True
False
True
True
cooling task
case: curve falling
Start Time
for case
False
setp.
setp.
tvac
True
1.5
-25
True
1.5
100
2
overshoot

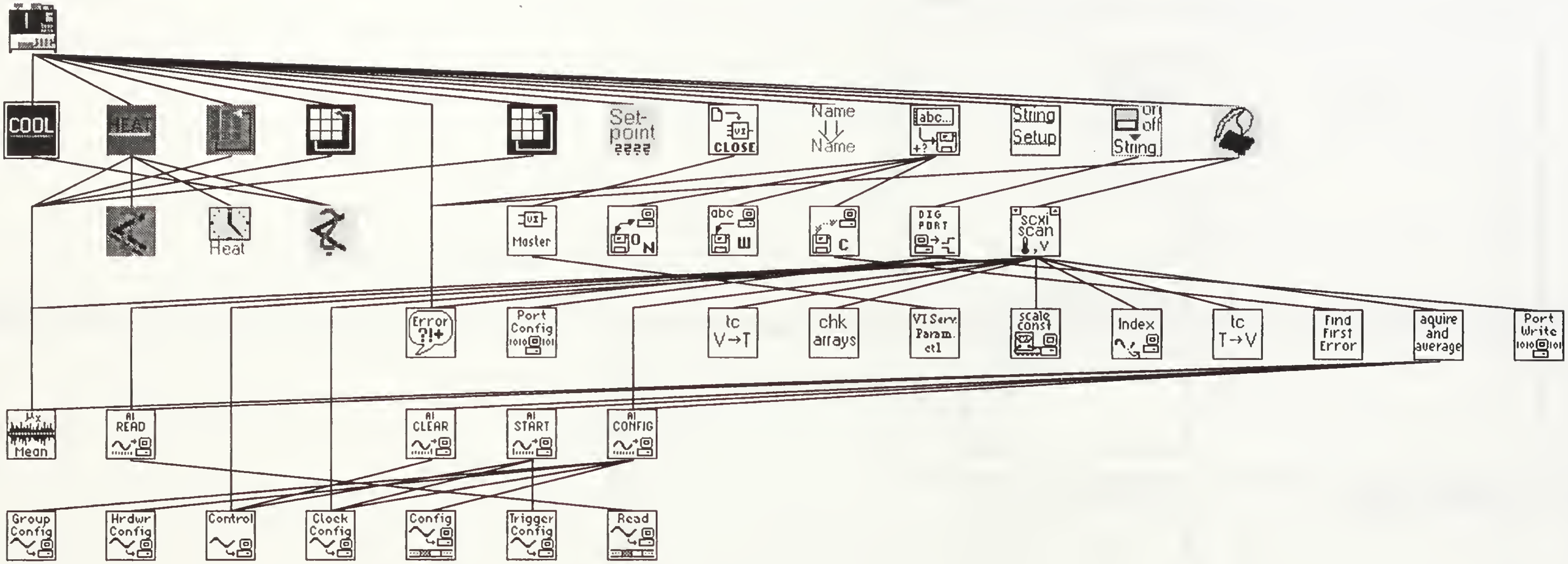

False
True
True
cooling task
case: curve falling
False
Start Time
for case
setp.
tvac
True
1.5
-25
True
1.5
100
2
=0
overshoot
False
2.5
True
1.5
False
not yet so close
> continue ambient
2


True
True
case: curve falling
Start Time
for case
False
True
1.5
100
2
overshoot
False
curve not falling
=> continue subzero !
2


True
case: curve falling
Start Time
for case
False
2
True
1.5
100
False
1
10
100
True
1.5
100
False
2
True
within time range
True
58



Position in Hierarchy


COOL
HEAT
Set-point
CLOSE
Name
Name
abc...
String Setup
on off
String
Heat
Master
abc
DIG PORT
SCXI scan
Error
Port Config
tc V→T
chk arrays
VI Serv. Param. ctl
scale const
Index
tc T→V
Find First Error
aquire and average
Port Write
Mean
AI READ
AI CLEAR
AI START
AI CONFIG
Group Config
Hrdwr Config
Control
Clock Config
Config
Trigger Config
Read

List of SubVIs

falling2.vi
C:\LABVIEW\EXAMPLES\DAQ\TVACUUM.LLB\falling2.vi

INITIAL DISTRIBUTION LIST

		No.Copies
1.	Dudley Knox Library, Code 052 Naval Postgraduate School Monterey, CA 93943-5002	2
2.	Chairman, (Code SP) Space Systems Academic Group Naval Postgraduate School Monterey, CA 93943-5000	2
3.	Jim Horning, (Code SP) Space Systems Academic Group Naval Postgraduate School Monterey, CA 93943-5000	1
4.	L. Todd Morris, (Code SP) Space Systems Academic Group Naval Postgraduate School Monterey, CA 93943-5000	1
5.	Dan Sakoda, (Code SP) Space Systems Academic Group Naval Postgraduate School Monterey, CA 93943-5000	1
6.	David Rigmaiden, (Code SP) Space Systems Academic Group Naval Postgraduate School Monterey, CA 93943-5000	1
7.	Michael Hengst, (Code SP) Space Systems Academic Group Naval Postgraduate School Monterey, CA 93943-5000	5